DOCUMENT RESUME

ED 190 081 IR 008 562

TITLE	Individualized Instruction for Data Access (IIDA). Quarterly Reports No. 5-6.
INSTITUTION	Drexel Univ., Philadelphia, Pa. Graduate School of Library Science.
SPONS AGENCY	National Science Foundation, Washington, D.C.
PUB DATE	79
GRANT	DSI-77-26524
NOTE	139p.; For related documents, see ED 145 826, ED 168 462, and ED 179 195, and also IR 008 561.
EDRS PRICE	MF01/PC06 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; *Computer Programs; Data Bases; *Individualized Instruction; *Information Retrieval; Online Systems; *Search Strategies
IDENTIFIERS	*Individualized Instruction for Data Access

ABSTRACT

These quarterly reports for June and September 1979 are two in a series concerned with describing a research project to design, test, and evaluate a machine intermediary which teaches and assists users of the DIALOG system. Report No. 5 explains in detail the instructional aspect of the Individualized Instruction for Data Access (IIDA) project and the benefits it is designed to impart to users of the system. Progress in testing the system with students is described as it relates to changes in the instructional program and general observations. The report illuminates the structure of the program, the contents of instructional exercises, and the current status of development and testing of IIDA, and provides a series of computer-generated exhibits. Report No. 6 has a more technical orientation, describing the present state of all of the computer programs that make up the IIDA system as well as their functions and operations. (FM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

ED190081

INDIVIDUALIZED INSTRUCTION FOR DATA ACCESS (IIDA)

Quarterly Report No. 5
June, 1979

Drexel University, School of Library
and Information Science
and the Franklin Research Center

NSF Grant No. DSI 77-26524

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Charles T. Meadow

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

IR008562

TABLE OF CONTENTS

1. OVERVIEW 1

2. STRUCTURE OF INSTRUCTIONAL PROGRAM 1

2.1 Introduction to IIDA 2
2.2 Choice of Exercises 3
2.3 Exercise 1 3
2.4 Exercise 2 3
2.5 Exercise 3 6

3. CONTENTS OF INSTRUCTIONAL EXERCISES 7

3.1 IIDA Teaching Philosophy 7
3.2 Exercise 1 7
3.3 Exercise 2 10
3.4 Exercise 3 11

3.4.1 Review of Basic Commands 12
3.4.2 Advanced Commands 12
3.4.3 Text Searching 12
3.4.4 Search Strategy 13
3.4.5 Database Descriptions 13
3.4.6 Beginning and Ending the Search 13
3.4.7 IIDA Facilities 13

4. CURRENT STATUS OF DEVELOPMENT AND TESTING OF IIDA 14

4.1 Meaning of Bibliographic Searching 14
4.2 Learing Goals 14
4.3 Time Considerations 14
4.4 Subject of Exercise I 15
4.5 Future Experiments 15

EXHIBITS

Exhibit 1 - Introduction to IIDA 1-1
Exhibit 2 - Menu of Exercises 2-1
Exhibit 3 - Exercise 1 3-1
Exhibit 4 - Exercise 2 4-1
Exhibit 5 - DIALOG Logon for Assistance Mode 5-1
Exhibit 6 - System Messages for Logon Failure 6-1
Exhibit 7 - Exercise 3 - Frames 1-8 7-1
Exhibit 8 - Exercise 3 - Detailed Instructional Frames 8-1

1. OVERVIEW

Individualized Instruction for Data Access (IIDA) is a research project involving the design, testing, and evaluation of a machine intermediary which teaches and assists users of the DIALOG system. The original intent of IIDA was to produce a fully operational service, a system that would be available to a variety of users on a self-supporting basis. However, when funding for the project was renewed in 1978 following the initial design study, emphasis on user training declined, the original goal was changed, and the project focussed on proving the concept of a search assistance system through testing a number of users on the system in an experimental setting. While the innovative aspect of IIDA remains its assistance mode, which helps the searcher in the conduct of his search, an instructional component provides the student a background on the use of DIALOG, and brings students with no previous instruction in searching to a level where they can perform a basic search and understand the messages issued by DIALOG and IIDA. The purpose of the instructional program is to teach the user to perform a search so that the effectiveness of IIDA assistance can be tested; the intent is not merely to demonstrate that searching can be taught by a CAI program.

The purpose of this report is to explain in some detail the instructional aspect of IIDA and the benefits it is designed to impart to users of the system. Progress to date, in terms of testing the system with students, will be described briefly as it relates to changes in the instructional program and general observations. Testing is scheduled to be completed in January 1980. Until then, conclusive results will be unavailable.

2. STRUCTURE OF INSTRUCTIONAL PROGRAM

The student's learning experience with IIDA is divided into three parts. Exercise 1 provides the student with an introduction to seven basic DIALOG commands and the IIDA HELP command. Exercise 2 provides an opportunity for the student to conduct a simple search, through DIALOG, using the commands introduced in Exercise 1. Exercise 3 presents further detail on the seven basic DIALOG commands and introduces the advanced commands which are used by experienced searchers. Exercise 3 includes details of how to use such DIALOG

facilities as SEARCH SAVE and text searching, and also includes detailed instruction on how to search the data bases available through IIDA and how to use IIDA facilities. An overview of various search strategies is also presented in Exercise 3. In the first and third exercises the student is required to enter particular commands and to view the system response to them; however, there are no tests of skill required of the student, as many CAI programs require for the student to proceed. The student is encouraged to gain a broad overview of DIALOG searching and an awareness of the help which IIDA can provide. It is intended that upon completion of both the instructional exercises the student will be able to perform a search in the assistance mode.

2.1 Introduction to IIDA

After completion of login procedures, the student is offered introductory material on the IIDA system. The purpose of this material, which consists of four frames of instruction, is to familiarize the student with IIDA and with procedures to be used in the conduct of all exercises and assisted searches. The learning goal for the student is that, on completion of the frames, he should be able to visualize the search process, use IIDA assistance facilities, correct typographical errors, and escape from any mode. He is given an explanation of the three instructional exercises and assistance mode.

Since the introductory material is an option which the student can review before beginning any exercise or prior to conducting an assisted search, it is not part of any single exercise. it serves not only the informative function, but also familiarizes the student with the use of the terminal and the mechanical procedures as well. He learns, for example, that he must enter a carriage return in order to send his message or command to the computer and that he can "erase" errors but in a different way from what he has experienced with a typewriter. The student may see any or all of the introductory material by answering "yes" to the question "would you like more information about IIDA?" which follows the first three introductory frames. Upon ending whatever frames of introduction the student selects, he is shown the options of Exercise 1, Exercise 2, Exercise 3 and Assistance Mode. It is

assumed that the student has learned enough in the introductory material to select one of the four options. Exhibit 1 is a transcript of IIDA's introduction.

2.2 Choice of Exercises

After completing the introductory frames or answering "no" to the invitation to see the introduction, the student selects the exercise he wants complete. He selects and enters either 1, 2, 3, or 4, the first three being Exercises and the fourth an assisted search. If the student does not want to proceed, he may enter the command /DONE to exit IIDA altogether.

One of the features of IIDA which demonstrates its versatility is the availability of choice in the order in which the student may proceed through the instructional sequence. A student may elect to ignore any of the exercises and proceed with an assisted search, he may follow the established order of IIDA exercises and then perform an assisted search, or he may proceed in any order. When the student exits from any exercise or completes a search in the assistance mode he is able to select again from the four options. Exhibit 2 illustrates the menu of exercises in IIDA.

2.3 Exercise 1

Exercise 1 presents the DIALOG commands BEGIN, EXPAND, SELECT, COMBINE, TYPE, PAGE, and LOGOFF within the context of a structured search. The student enters commands, sees the system's response to the commands, and receives a brief, very basic explanation of what has been seen. Completion of Exercise 1 takes approximately 20 minutes.

The present Exercise 1 simulates a search in COMPENDEX (file 8) on the topic of the use of solar collectors or heat pumps in residential buildings. A complete transcript of Exercise 1 is reproduced in Exhibit 3.

2.4 Exercise 2

Exercise 2 gives the student the opportunity to conduct a real search in COMPENDEX. Two search topics, both appropriate to COMPENDEX, are suggested to the student, although he may actually search on the topic of his choice.

The suggested search topics are:

1. The use of asphalt cement for paving roads and bridges

2. The use of passive solar energy to heat residential buildings

The topics were selected for several reasons: first, these topics have been in the news enough that most engineers or engineering students will know something about one or both topics. This will enable the IIDA student to select appropriate search terms without much difficulty. Second, the data bases contain a large number of items on these topics, and so the student is assured at least moderate success if he uses only the simplest search strategy and the basic commands which were introduced in Exercise 1.

Upon seeing the topics suggested for Exercise 2, the student is asked to establish a goal in terms of the number of citations he hopes to retrieve. He is told that he can change his goal at any time during the search and is instructed on how to do it. The student is asked to take a moment to think of appropriate terms related to the topic he has chosen; the setting of the goal and the "moment" to think of search terms contribute to the impression the student receives about the process of bibliographic searching as a whole. IIDA tries to instill in the student the idea that he should not begin a search without establishing a goal and without thinking through at least part of a search strategy. While the student will receive more detailed instruction on logical search strategies in Exercise 3, he may begin to form the habit of thinking about his goal, based on how broad or detailed a list of citations he needs, and the terms which he can use in his search, by going through the process rather than being instructed on the process. The preliminary material to the actual conduct of the search in Exercise 2 is illustrated by Exhibit 4.

Following the preliminary material to Exercise 2, the student waits a matter of a few seconds for the logging in procedure to be completed with DIALOG. IIDA tells the student that IIDA will place the phone call to DIALOG, and that a short delay is to be expected. He is informed of the initialization and status of all telecommunications processes, when the DIALOG login procedure is begun, and when he is connected to DIALOG. If the call

cannot be completed through TYMNET, and TELENET is used for a second attempt, or vice versa, and the student is so informed. This procedure, of keeping the student informed of what is happening, even though he may not be directly involved in the process, is designed to maintain his interest and to keep him assured that he needs do nothing but wait. This reassurance is particularly helpful, it appears, to those students who have not experienced online delays previously. A transcript of a typical user's view of the DIALOG login procedure through IIDA is illustrated in Exhibit 5.

It is important to note that if, for any reason, the DIALOG system cannot be accessed, the student is informed and is given the information that "IIDA will quit for now." This signals that the student should log out and try again later. Once again, the student is assured that he has done nothing "wrong". This is in keeping with the IIDA practice of letting the student know what is happening each step of the way and letting the student have every opportunity for success with the system, with as little frustration as possible. Exhibit 6 illustrates the messages received by the student following unsuccessful attempts to connect with the DIALOG retrieval system.

Although Exercise 2, following the preliminary information regarding suggested seach topics, operates identically to assistance mode, the Exercise is still instructional without taking the instructional frame format used in Exercises 1 and 3. Here the student "learns by doing" an actual search.

The search is conducted in a usual manner, following the IIDA instruction to begin the search with the BEGIN command. The student is informed by IIDA of syntax and strategy errors, and can access the help program at any point in the search simply by entering /HELP. IIDA asks the student to rate the relevance of citations TYPED out. This rating is used for several purposes: first, IIDA converses with the student on his progress in terms of relevance. Secondly, IIDA can inform the student that his search may be nearly complete, when he has TYPEd a fair number of highly relevant citations. Finally, the rating of the relevance of individual citations forces the student to think about what citations he has retrieved and to look for concepts that he has not included in his search.

Upon completion of Exercise 2, the student should have a good understanding of the basic seven commands introduced in Exercise 1, a sense of confidence in terms of what kind of response to expect of the system in its responses to the basic commands, and a beginning notion of search strategy. While the idea of a logical search strategy has not yet been introduced by instruction, the student has had the opportunity to "discover" what seems to work and what does not. In any event, the student knows that more instruction awaits him in Exercise 3, and while he may have seen a good bit of Exercise 3 using/HELP, he will now be able to proceed through as much of Exercise 3 as he needs to review basic commands and to learn the more advanced techniques.

Completion time for Exercise 2 cannot be predetermined as each student begins it with a different degree of expertise in using the material introduced in Exercise 1. For purposes of testing the system with students, one-half hour was scheduled as a minimum, although rates of completion thus far have varied from 15 minutes to over one hour.

2.5 Exercise 3

Exercise 3, the longest exercise of IIDA, serves two purposes. First, it is an instructional program, similar to Exercise 1, in that the student can follow through frames, in sequential order, learning new commands and techniques and practicing use of the material while learning. Secondly, the frames of instructional material are used as the /HELP library; the material can be accessed by the student during any exercise or in the course of an assisted search. Through a series of eight frames the student can be introduced to the entire contents of the exercise, and on completion of the eight frames he may follow any of seven lines of detail. If the student elects to see only the eight introductory frames he may use the eight frames as a table of contents and access needed inforamtion later, during a real assisted search. If the student does not have the transcript of the first eight frames of the exercise with him, or if he elects not to use Exercise 3 as instructional material, he may still access it though the /HELP library, through a series of frames and menus, presented when he enters /HELP during a search.

The completion of the eight frames which introduce the contents of Exercise 3 takes the student approximately 10 minutes. If the student elects to continue in the instructional mode, he may take up to an hour or more to follow all seven lines of discussion. The material goes into considerable detail, but after each frame the student has the option of going into more detail, selecting another subject within the Exercise, or exiting the Exercise altogether. Allowing the student to determine the level of detail he wishes to pursue is in keeping the overall IIDA philosophy of having the student learn at his own pace, through self discovery. Exhibit 7 is a transcript of the first eight frames of Exercise 3. Exhibit 8 illustrates all instructional units of Exercise 3 in sequential order.

3. CONTENTS OF INSTRUCTIONAL EXERCISES

3.1 IIDA teaching philosophy

The instructional programs of IIDA were designed to allow the student to learn through self discovery. Instead of presenting commands, their meanings, and then an opportunity for the student to use what he has learned, the IIDA teaching exercises were developed to allow the student to use the commands first, to see the system response to the command given in the context of a search, and finally to see a very brief explanation of what happened. It was felt that this approach would allow the student to gain an understanding of the whole process of searching rather than the narrower view of just the DIALOG commands.

3.2 Exercise 1

Exercise 1 consists of instruction on basic DIALOG commands. It begins with a brief explanation of the exercise, explaining that instruction will take place in the context of a simulated DIALOG search. The topic of the search is introduced so that the student has enough background to carry out the search.

IIDA issues instructions to the student to enter certain commands, starting with the BEGIN command. Upon entering the IIDA directed command BEGIN 8 the student sees the exact response that he would see if he were doing an unassisted DIALOG search. Following the system response, the student sees an explanation of what his command has evoked from the system. In Exercise 1, the explanations are extremely brief, running an average of ten lines. They tell the students the most basic facts about the command and the system response. More detailed explanations, introductions to abbreviations for the commands, and alternative ways of approaching each step of the search are saved for Exercise 3.

The student proceeds with the search by selecting the term "solar collectors" using the SELECT command. An explanation of the SELECT command and response follows, and the idea of looking at some of the records retrieved is introduced. The student is then told to enter the TYPE command, and he sees two records.

Next, the notion of finding additional terms to express the subject of the search is introduced, and the student is told to enter EXPAND HEAT PUMPS. The resultant EXPAND table is explained, and in the same context the PAGE command is presented.

The student is next instructed to SELECT HEAT PUMPS and SELECT HEAT PUMP SYSTEMS. The idea of combining the concepts SELECTed into a single set is suggested and the student is told to enter COMBINE 1 OR 2 OR 3. After the response, the student is given an explanation of the response and of the use of the OR logical operator to create the union of several sets.

The program then instructs the student to add the concept of "residential" to his search; the student is told to SELECT RESIDENTIAL and to COMBINE the concept with the previous set using the COMBINE command. The use of the AND logical operator is then introduced. Several records are TYPEd in another format (format 5) which reveals to the student the fact that abstracts are available and can be useful in determining the usefulness of a specific item.

The IIDA /HELP command and its use is introduced so that the student may review the sets created in the search. By using the /HELP command, the student also learns what other kinds of assistance are available through IIDA, and he may begin to anticipate using /HELP in later exercises and actual searches.

Finally, the LOGOFF command is introduced and the student ends his first search experience.

IIDA then gives the student an eight-line review of Exercise 1 in the form of a summary of commands introduced.

The content of Exercise 1 follows the objectives of the entire IIDA teaching philosophy. First, the instructions are as brief as possible. The student is given the chance to learn by doing, rather than learning from a textbook approach. Secondly, the student sees results of his efforts immediately. If he makes an error, IIDA issues the message to "try again entering (command indicated) exactly." The continuous feedback enables the student to see his errors and correct them easily. It also teaches the student the necessity for following the instructions exactly as issued; this aspect will become even more important as the student learns additional commands and techniques in later instructional units and will also serve the student well throughout the search experience. Third, the language of Exercise 1 avoids information retrieval jargon. Terms such as "boolean operators" and the like are avoided where possible and, where necessary, terms are explained briefly. Fourth, and finally, the instruction avoids talking down to the student and at the same time attempts to keep the tone friendly but not "cute." It is the view of the project staff that students who undertake the program of instruction in online bibliographic searching are motivated to learn something from which they can benefit. Hence, humorous responses to errors and condescending affirmations are not used in IIDA.

Upon completion of Exercise 1 the student is invited to continue IIDA with Exercise 2, Exercise 3, or an assisted search.

3.3 Exercise 2

Exercise 2 is not a tutorial exercise in the sense that Exercises 1 and 3 are. Exercise 2 provides a chance for the student to perform a search using the commands and strategy learned in Exercise 1. Exercise 2 directs the student to search on one of two suggested topics, but in actuality the student may search on any topic of his choice. The topics suggested were designed to assure the student a fair degree of success at searching because of the number of items on the topics in the data base and because the topics are familiar to most people who read a daily newspaper or newsmagazine.

The student selects his search topic and is informed of the data base to select to do his search. Both searches should be performed in COMPENDEX. It is suggested that the student set a search goal for a specific number of citations, and then he is further encouraged to think about useful search terms for a few moments. This is the student's "learning by doing" basic lesson in search strategy; the lesson will be given in greater detail in Exercise 3.

The student uses the commands introduced in Exercise 1 to perform his search. There are, however, a few restrictions: the student is prohibited in typing excessive numbers of citations; he is informed that, for example, four citations in format 5 are sufficient to view a set. The student may also be recommended to use the /HELP facility for assistance. This is based on the fact that the student's progress with the search is monitored by IIDA for both syntax and strategy problems. When the student makes mistakes, IIDA instructs the student as to his error and often suggests that he use /HELP. IIDA's advice may be used or ignored, but the student nevertheless becomes familiar which more of the options IIDA offers in the way of assistance. IIDA suggests specific frames in /HELP for review on syntax or strategy.

Thus, Exercise 2 instructs the student in two ways: by allowing the student to do his own search and to practice what was learned and by offering assistance during the course of the search thereby teaching the student more about the assistance available though IIDA.

In Exercise 2 a limited number of commands may be used. In Exercise 3, additional commands and techniques are introduced and become part of the student's repertoire.

3.4 Exercise 3

Exercise 3, taken as an instructional exercise rather than as part of /HELP, is structured in a reference book format. The student first sees eight frames which introduce the topics covered in the remaining 80 frames. An explanation of the contents of seven major sections of instruction is presented. The major sections are:

Review of Basic Commands
Advanced Commands
Text Searching
Search Strategy
Database Descriptions
Beginning and Ending a Search
IIDA Facilities

A transcript of the eight introductory frames for Exercise 3 is included as Exhibit 7.

The eight introductory frames are seen by the student in sequential order. Following the eight frames, the student is given the choice of continuing with instruction by entering a two-digit frame number for the frame of instruction he desires or of skipping instruction altogether and going on to an assisted search.

If the student elects to continue with the instruction, he may select any two digit frame number to begin his instruction. From that point he may continue in as much or as little detail as he wishes. When following a particular line of discussion in Exercise 3 the student is occasionally asked to enter specific commands and is then able to view the system response to those commands introduced. The various commands and responses do not comprise a complete search as they do in Exercise 1.

The contents of Exercise 3 is summarized in the following section. A complete transcript of the final 80 frames of Exercise 3 is included as Exhibit 8.

3.4.1. Review of Basic Commands.

The basic commands introduced in Exercise 1 are reviewed. The student is given additional information on BEGIN, EXPAND, SELECT, COMBINE, TYPE, and PAGE commands. The abbreviations for these basic commands are introduced and the student is told that he can receive even more information on the full BEGIN and use of BEGIN command to change files. The more detailed information is reserved for his use when he calls the /HELP facility during a search. New information is given on selecting from EXPAND tables by using E-reference numbers, using logical operators OR and NOT, and selecting the appropriate TYPE format for viewing items retrieved.

3.4.2 Advanced Commands.

Advanced commands and techniques are introduced for the first time. Explanations are presented for file selection, limiting, the use of TYPE and PRINT commands, abbreviations for commands and logical operators, stacking, use of the set range short cut, use of the DISPLAY SETS command, and use of DIALOG's SEARCH SAVE feature. The student's knowledge of the BEGIN command is expanded to include use of the command to change files. The FILE command is introduced. LIMIT and LIMITALL commands are introduced and the student is informed of the availability of more detail on limiting to specific dates and using multiple criteria with the LIMIT commands. A table of abbreviations of commands, with examples of their use, is presented, and the student has an opportunity to practice the use of time saving techniques of stacking and use of ranges with the COMBINE command. The DIALOG feature SEARCH SAVE is introduced for future reference.

3.4.3. Text Searching.

Upon completion of instruction on basic and advanced commands and advanced techniques, the student is introduced to the concept of text searching. Considerable detail is presented on the text searching techniques with opportunities for the student to practice. Truncation, the use of prefixes and suffixes, and the use of infixes are introduced and the student practices such techniques as expanding using the AU= prefix and searching the

title field using the /TI suffix. The infix symbols are introduced and a table is presented on the use of each. Students are alerted in this section to data base dependent practices in text searching, and the fact that searchable fields vary is explained. This prepares the student for future instructional units on the individual data bases.

3.4.4. Search Strategy.

Definition of the search topic in advance of going online, the analysis of the search topic and the division of the topic into facets, use of short cuts, and the importance of a search goal are stressed in the exercise. The student is informed that more detail is available, during a search, on specific strategy problems such as null set generation, cycles of commands, and the formulation of logical strategies.

3.4.5. Database Descriptions.

This group of frames tells the student about each of the IIDA databases. ERIC, NTIS, COMPENDEX, and ONTAP ERIC are first discussed generally, as their contents and coverage will affect the student's choice for searching. The student is informed of database dependent search methods and formats and is invited to access the information through the help facility.

3.4.6. Beginning and Ending the Search.

This group of frames reviews the login procedure for IIDA, and would enable the student to begin his search, from the initial dialing to the telecommunications network, on his own. DIALOG and IIDA prompts are explained, the LOGOFF command is explained, and system responses to BEGIN and LOGOFF are discussed. The student is informed of the availability of time and cost messages during the search.

3.4.7. IIDA Facilities.

This section introduces the IIDA diagnostic facility which is part of the assistance mode, and discusses the /HELP facilities. Detailed accounts of each are available in the more detailed frames available during the search.

4. CURRENT STATUS OF DEVELOPMENT AND TESTING OF IIDA

The initial users of the IIDA system were engineering undergraduate students at Drexel. Some 30 students enrolled in a technical writing course participated in an experiment to test IIDA as a teacher of online bibliographic searching. Students completed Exercises 1, 2, and 3, and then were given the opportunity to do assisted searches on the topic for which they were doing research for a term project. While the performance of the students as searchers is currently being evaluated, and the results of the experiment will be reported in a later Quarterly Report, some informal observations are in order here.

4.1. The Meaning of Bibliographic Searching

The initial students to use the system seemed to be working under a misconception that they would retrieve the actual facts about their search topic rather than citations. The instructional program was revised to reflect this and now contains a more thorough explanation of online bibliographic searching than the IIDA staff originally believed to be necessary.

4.2. Learning Goals

The first few students to receive IIDA training did not seem to realize that they were to learn the basic commands presented in Exercise 1, but appeared to expect that in Exercise 2 and subsequent searches they would be told precisely what to do. After only a half dozen subjects had completed the first two exercises, IIDA staff rewrote the introductory material to Exercise 1 and indicated in that introduction that the student would be expected to use what he had learned in subsequent exercises. The current Exercise 1, reflected in this report, stresses the importance of the basic commands and in fact offers a short review of the commands at the end of the exercise.

4.3. Time Considerations

The original program allowed the student 90 seconds from the end of the system response to enter a command. After 90 seconds, IIDA issued a warning that the student would be logged off if he did not issue a command. Many

students complained that that did not allow enough time to read and digest the instructional material or, during a search, to decide what to do. The more "computer literate" subjects soon learned to simply enter a carriage return to satisfy the requirement. Others felt impelled to issue a command and reported their feelings of time pressure in the user questionnaires. As a result, the time limit was extended to 3 minutes. The rate and frequency of complaints on this feature immediately subsided.

4.4. Subject of Exercise 1

The students who first tested IIDA used the initial version of Exercise 1, which taught the basic commands in the context of a search in ERIC (File 1) on the subject of automated library circulation systems. After a few subjects had completed the Exercise, informal conversation and comments made by the subjects revealed that the topic was of so little interest to them that they did not understand why such a topic was used. There was also some confusion about the concept of "library." Apparently some misinterpreted the EXPAND command, and thought that to see the alphabetical listing of terms, they should enter EXPAND LIBRARY no matter what term was to be EXPANDed. It was after the initial subjects had completed the exercise that IIDA staff decided to rewrite the exercise in COMPENDEX and to use a subject appropriate to that file, the use of solar energy for heating residential buildings.

4.5. Future Experiments

Plans for evaluation of the experiments undertaken at Drexel are discussed in Quarterly Report #3. That report also gives more detail on other evaluations to be used.

At the time this report is written the final phase of the evaluation task is being conducted in the offices of Exxon Research and Engineering Company, which has lent its facilities and recruited a large number of engineering personnel to serve as test subjects. Some program revisions were made prior to the start of this evaluation, based upon earlier feedback from subjects. While no firm data is yet available, informal indications are that the Exxon professional staff are performing better and liking the system better than the Drexel undergraduate students. More detail and analysis of this experiment will be presented in Quarterly Report Number 8, in March 1980.

EXHIBIT 1 -- Introduction to IIDA

Welcome to IIDA. Would you like a brief introduction to the use of the system? (y or n)

I? y

The system you are working with is called IIDA. Its purpose is to teach you how to perform computer assisted searches of bibliographic files and to help you to do the searches.

The files you search contain title, author, subject and other such descriptive terms, and often have an abstract. These records usually describe published articles or reports and sometimes books.

What you retrieve from a search is a list of citations and abstracts, not the articles themselves. We're sorry, but you still have to order the articles from your library.

You will be using a search system called DIALOG. DIALOG is a computer program that takes your instructions and uses them to guide a search of its files.

DIALOG has nearly 100 files, or data bases, available for searching. We will be limited to just one of these for the purpose of this exercise. The single file to be searched is COMPENDEX, a file produced by the Engineering Index, Inc., covering published information in all fields of engineering.

IIDA is not the search system. IIDA is a system that teaches how to use DIALOG and then helps you to do so. After you have completed your training, you could search DIALOG directly, on your own, or use IIDA to help you. For experimental purposes, we will ask you to continue using IIDA for your searches.

Would you like more information about IIDA?

I? y

IIDA consists of three training exercises and an assistance mode. The training exercises are:

1. Basic search commands: the 7 basic commands that you need to perform a search;
2. A sample search, using each of the basic commands;
3. Advanced commands and search techniques: more commands and ideas on how to use them. You can learn these at your own speed. They are optional.

The assistance mode is the program that helps you do a search on your own. It can be used independently of the training exercises.

/-/

Before we get to the commands that you use for telling DIALOG how to do a search, let us cover a few points about how to deal with IIDA.

Any time you need help, use the command /help. You will be given access to a library of information about using DIALOG and IIDA.

Sometimes, instead of looking up some information through /help, you just want to know what to do next. You can get some advice by asking for 'quick advice' or saying to IIDA /qa. The advice you get will be a general suggestion about how to proceed. If you need a definition of a command, or list of commands available, or other information, ask for /help.

Would you like still more information about IIDA?

I? y

If you have entered a command and realize you've made a mistake, you can cancel it by typing @. As soon as you've typed the @ you can begin entering your new line of type.

If you want to erase just one or a few characters that you've typed, enter a #, once for each character you want to erase, from right to left. If you've typed IIDX you can change it to IIDA by typing #A. The entire line would look like IIDX#A, but the DIALOG computer would see only IIDA.

To transmit a message, a line or a command to the computer, always type a carriage return (button marked RETURN). This is a signal that your message is complete. We use the abbreviation CR for carriage return. If your message is /help, you type /helpCR.

When you are talking to DIALOG, and the computer is waiting for you to give a command, it will prompt you with the symbols D?. When IIDA wants you to provide some input, it says I?. It is your turn to send a message only after you've received one of these prompts.

Finally, whenever you are in one mode or exercise of IIDA and you want to exit from it you can reply /done to a prompt from either IIDA or DIALOG. So, if you feel you've had enough instruction, or you want to take a break, type /done.

Now, let's start the exercises.

EXHIBIT 2 -- Menu of Exercises

```
To begin one of the following enter its number:
          Exercise 1: basic DIALOG commands
          Exercise 2: practice DIALOG search
          Exercise 3: advanced commands and search strategy
                   4: search assistance only
Enter '/done' if you are finished with IIDA.

I? 1
```

EXHIBIT 3 -- Exercise 1

The purpose of doing a search through a system such as DIALOG is to get information quickly about what has been published on some subject that you specify. The high speed of searching by computer gives you quick response and this enables you to try more than one approach to defining your subject, which you may find necessary if you can't get what you want right away.

The records you retrieve are something like library catalog cards, but usually have an abstract with them, to give you more information about the article or book they describe.

The subject of the article is conveyed by a group of words or phrases called DESCRIPTORS. These are assigned to the record by indexers, who are trained in both the subject matter and in handling bibliographic records.

You will learn in exercise 3 that it is also possible to search a record by searching for the occurrence of particular words in a title or abstract or descriptor field.

You conduct a search by sending to DIALOG a series of commands telling DIALOG what to search for, or what records to type out to you on your terminal. IIDA contributes by teaching you the commands, providing advice on what to do next, and diagnosing your performance, trying to see if your sequence of commands shows signs of not leading to a useful result.

If IIDA detects a problem in the way you are conducting your search, it will tell you so, and you can act upon its messages or ignore them.

The first thing you do in a search is tell DIALOG what data base you want to search. One way to do this is with the BEGIN command. There is another way and there are other functions of BEGIN, but we can ignore them for now.

The BEGIN command, like all DIALOG commands, consists of a command and an argument. The argument is the number of the data base you want to search.

We're now going to take you through a simple search. The subject of the search is the use of solar collectors or heat pumps in residential buildings. You'll be told what to type. Type in exactly what you're told and then you'll see what DIALOG does with that command. Later on, you can do a search on your own.

Enter BEGIN 8

```
D? begin 8
        21nov79 11:40:51 User9002
  $0.18  0.012 Hrs File1*
File8:COMPENDEX 70-79/AUG
(copr. Engineering Index Inc.)
        Set Items Description (+=OR;*=AND;-=NOT)
        --- ----- ---------------------------
```

You may have gotten more than you bargained for here. The BEGIN command results in some administrative information from DIALOG. It tells you the time and date of your command, the amount of money you've spent until now, what data base or file you're working with or are 'in', and some other news that we suggest you ignore for now.

Normally, what you would see next would be a prompt symbol (D?) from DIALOG. We've interrupted for a moment to tell you what is coming next.

The next thing you probably want to do is ask DIALOG to search for some subject terms. Suppose you want to try finding what is available in COMPENDEX on solar collectors.

The command is SELECT, that is, DIALOG will select all the records having this subject term.

Enter SELECT SOLAR COLLECTORS

```
D? select solar collectors
        1      4 SOLAR COLLECTORS
```

What this means is that DIALOG has selected the records that include your search term in them. The set of records retrieved is set number 1. This set has 4 records in it. DIALOG then repeats your set-defining search term just as a reminder to you.

To review, you asked DIALOG to retrieve records containing the term solar collectors. It did this and then told you it had created set no. 1, that this set had 4 records in it, and reminded you of the search term.

Knowing how many records are in the set is interesting, but it doesn't solve your problem. You'd probably like to see some of these records, to be sure they have what you want in them.

If you want to look at records, you have to tell DIAoLOG to type them for you. The command is TYPE.

It has a complicated argument that includes the set number to be typed, a code for what format that record is to be typed in, and what records are to be typed.

Enter TYPE 1/2/1-2

D? type 1/2/1-2

1/2/1
ID NO.- EI790866454 966454
SOLAR COLLECTOR PERFORMANCE: A DEPENDENCE UPON COATINGS.
Wolf, Richard E.
DeSoto, Inc, Des Plaines, Ill
J Coat Technol v 51 n 652 May 1979 p 49-53 CODEN: JCTEDL
DESCRIPTORS: (*WATER HEATERS, *Solar), (SOLAR RADIATION, Absorption), (PROTECTIVE COATINGS, Physical Chemistry), CARBON BLACK,
IDENTIFIERS: BLACK COATINGS, SOLAR COLLECTORS
CARD ALERT: 643, 539, 647, 801, 802, 804

1/2/2
ID NO.- EI790534306 934306
DYNAMIC MODELING OF WATER- AND AIR-HEATING SOLAR COLLECTORS.
Savery, C. William; Hofmann, Mark
Drexel Univ, Philadelphia, Pa
Int Heat Transfer Conf, 6th, Toronto, Ont, Aug 7-11 1978 Publ by Natl Res Counc of Can, Toronto, Ont, 1978. Available from Hemisphere Publ Corp, Washington, DC v 2 p 147-151
DESCRIPTORS: (*HEAT TRANSFER, *Radiation), SOLAR CELLS,
IDENTIFIERS: SOLAR COLLECTORS, DYNAMIC MODELS
CARD ALERT: 641, 702

What you just saw was the first two records of your set typed in what we call format 2. It looks something like a library catalog card, and is meant to. You can see the title, author's name, source (i.e., what journal it was published in), dates, and so on. These records don't include abstracts. To get abstracts you use a different format number. The type outs are longer. We'll show you one later.

If you'd been really lucky, your search might have ended right here. But these records don't seem to be the best answer to our original question (which was the use of solar collectors or heat pumps in residential buildings).

Maybe we should consider some other ways of expressing our subject. For example, do we search for heat pump, heat pumpS, or what? It would be nice if we had a dictionary to look in, to see what terms are to be found in the DIALOG files. We do have one and we search it with a command called EXPAND.

Enter EXPAND HEAT PUMPS

```
D?  expand heat pumps
Ref   Index-term                Type Items RT
E1    HEAT PUMP DEHUMIDIFIER--      1
E2    HEAT PUMP PERFORMANCE---      2
E3    HEAT PUMP PROCESS-------      1
E4    HEAT PUMP STATIONS------      2
E5    HEAT PUMP SYSTEMS-------    515
E6   -HEAT PUMPS--------------      6
E7    HEAT RADIATION----------      3
E8    HEAT RANSFER------------      4
E9    HEAT RATE---------------      1
E10   HEAT RATE TEST----------      1
E11   HEAT RECLAIM------------      1
E12   HEAT RECOVERY-----------     43
E13   HEAT RECOVERY BOILERS---      3
E14   HEAT RECOVERY EQUIPMENT-      1
E15   HEAT RECOVERY EXCHANGER-      1
E16   HEAT RECOVERY EXCHANGERS     1
E17   HEAT RECOVERY
        INCINERATION SYSTEMS--      1
E18   HEAT RECOVERY METHODS---      2
E19   HEAT RECOVERY NETWORKS--      1
                                 -more-
```

You have retrieved a segment of an alphabetically-ordered list of terms--all the terms used in COMPENDEX records. The word you searched on is on line 6, marked with the hyphen before it.

The left-hand column gives a line number on this display. E is for expand. E1 is line one of the expand display.

The right-hand column tells you how many records in COMPENDEX contain this term. We can see that heat pumps occurs in 6 records, and heat pump systems occurs in 515.

From this display we can see that it will be worthwhile searching on heat pump systems as well as heat pumps.

We can also see that there may be more useful terms that did not appear on this display. Whenever you see the notation -more- on a DIALOG display, you can give the command PAGE to see the next 'page'.

Enter PAGE

```
D?  page
Ref  Index-term                Type Items RT
E20  HEAT RECOVERY STEAM
       GENERATORS------------           1
E21  HEAT RECOVERY SYSTEMS---          10
E22  HEAT RECOVERY WHEELS----           2
E23  HEAT RECUPERATOR--------           1
E24  HEAT RECUPERATORS-------           1
E25  HEAT REGENERATORS-------           1
E26  HEAT REJECTION----------           2
E27  HEAT REJECTION CONTROL--           1
E28  HEAT REJECTION SYSTEM---           2
E29  HEAT REJECTION SYSTEMS--           1
E30  HEAT RELEASE------------           6
E31  HEAT RELEASE RATE-------           2
E32  HEAT RELEASE RATES------           1
E33  HEAT REMOVAL TROFFERS---           1
E34  HEAT REQUIREMENTS-------           1
E35  HEAT REQUIREMENTS OF
       ROOMS-----------------           1
E36  HEAT RESISTANCE---------         375
E37  HEAT RESISTANT ALLOY----           1
                                     -more-
```

The PAGE command has given us a continuation of the expand display. As long as we see -more- we could continue to PAGE, but this seems to be enough. We don't see any new terms we'd like to use.

Let's retrieve records containing heat pumps or heat pump systems. To do this, first SELECT each of these terms.

Enter SELECT HEAT PUMPS

```
D?  select heat pumps
          2      6 HEAT PUMPS
```

Enter SELECT HEAT PUMP SYSTEMS

```
D?  select heat pump systems
          3    515 HEAT PUMP SYSTEMS
```

Now you have created two more sets, numbers 2 and 3. Note that DIALOG numbers them for you.

It would now seem useful to form a single set that contained either solar collectors or heat pumps or heat pump systems.

Enter COMBINE 1 OR 2 OR 3

D? combine 1 or 2 or 3
4 522 1 OR 2 OR 3

Set 4 contains 522 records. A record in set 4 has either the term solar collectors or heat pumps or heat pump systems in it. Set 4 is called the union of sets 1, 2 and 3. The logical operator for creating the union of two or more sets is OR.

Suppose we want records which are in set 4, but also mention the term residential. Again, we have to SELECT first.

Enter SELECT RESIDENTIAL

D? select residential
5 1168 RESIDENTIAL

Remember, we wanted to create a set of records that are on both the subject of the energy devices and on residential applications.

Think for a moment how you might do this.

Here is how it is done in DIALOG.

Enter COMBINE 4 AND 5

D? combine 4 and 5
6 51 4 AND 5

Set 6 has 51 records in it. It is called the intersection of sets 4 and 5.

Let's look at a record of set 6.

Enter TYPE 6/5/1

D? type 6/5/1
6/5/1
ID NO.- EI790861683 961683
NIEDERTEMPERATURFLAECHENHEIZUNG IN NUTZBEREICH NEUER ALTERNATIVENERGIEN AUS SONNE, LUFT, ERDE UND WASSER. $left bracket$ Low-Temperature Heating Using Sun, Atmosphere and Ground as Source of Heat $right bracket$.
Lindner, Helmut
Abteilung Forsch-Entwicklung-Anwendungstech Leverkusen, Ger
Elektrowaerme Int Ed A v 36 n A4 Jul 1978 p A224-A227 CODEN: EIEADN

Low-temperature heating in the range from $plus$ 50 to $plus$ 25 and technical limitations are known. The use of heat pumps recovering heat from several environmental sources for space-heating purposes is part of the efforts made to save energy and to reduce the costs of heating. Low-temperature radiators with a large convective area are already available and floor heating systems using low-temperature water offfer a number of advantages. Such large-surface heating systems must, however, be designed, dimensioned and installed with great care. The system cannot only be used for space heating in new residential buildings but can also be used in existing buildings. In German.
DESCRIPTORS: *HEAT PUMP SYSTEMS, SOLAR ENERGY,

We used format 5 this time and got an abstract as well as the citation. We also specified that we wanted only one record typed. The abstract is useful, but time consuming. You have to learn to use your judgment about which format to use.

At this point, you might decide you got what you wanted and end the search. Before you do, though, you might want to review the sets you've created just to be sure you used the right terms and combinatorial logic. This will give us a chance to show you how IIDA's help program works.

Enter /HELP

D? /hi@/help
HELP is available on the following:
1. Definitions of search commands and techniques
2. Advice on current problems, if any
3. Commands given in this search
4. Sets created in this search
5. Records viewed -- a summary
6. Errors made -- a summary
7. Descriptors used -- a summary
8. Change to another exercise/mode

(Note that it's ok to give an IIDA command following a DIALOG prompt).

Enter 4 to review the sets you've created.

D? 4

```
    SET  SET       AVG.
CMD NUM  SIZE REFS REL. COMMAND EXTENDED ARGUMENT
--- ---  ---- ---- ---- ------- ----------------

 2   1      4   2  0.00 select  solar collectors
 6   2      6   1  0.00 select  heat pumps
 7   3    515   1  0.00 select  heat pump systems
 8   4    522   1  0.00 combine (solar collectors) or (heat pumps) or
                                  (heat pump systems)
 9   5   1168   1  0.00 select  residential
10   6     51   1  0.00 combine ((solar collectors) or (heat pumps) or
                                  (heat pump systems)) and residential
```

This shows you the set numbers, set sizes, commands used, and the arguments that defined the sets. Ignore the other columns for now.

The help program asks you what you want to do next. Suppose you want to go back to your search. You can exit from help and return to the search all at once by giving your next DIALOG command.

Assuming you were satisfied with this search, you could tell DIALOG you are done by giving its search terminating command LOGOFF.
Enter LOGOFF when prompted.

```
D?  logoff
        21nov79 11:52:55 User9002
  $2.93  0.195 Hrs File8 4 Descriptors
```

You've now completed a search. DIALOG has logged you off and ended with a statement of the time, date, and cost of your search.

You've also completed IIDA's exercise 1. You should be ready to do a search on your own. Remember a few points. The commands we introduced were:

```
     BEGIN to start and choose a file
     SELECT to retrieve records containing a term
     TYPE to have DIALOG type the records on your terminal
     COMBINE to form new sets by combining existing ones
     EXPAND to look in DIALOG's dictionary
     PAGE to see the next page of the dictionary
     LOGOFF to end a search
     /HELP to get IIDA's assistance.


Terminating Exercise 1.
To begin one of the following enter its number:
          Exercise 1: basic DIALOG commands
          Exercise 2: practice DIALOG search
          Exercise 3: advanced commands and search strategy
                   4: search assistance only
Enter '/done' if you are finished with IIDA.

I? 2
```

EXHIBIT 4 -- Exercise 2

Exercise 2 allows you to practice searching on DIALOG. It is possible to perform a successful search on a rather specific topic using only the six basic commands learned in Exercise 1. Practice your search skills by finding citations on one of the following topics using COMPENDEX (file 8):

1. Asphalt paving for road and bridge surfaces
2. Passive solar energy used to heat residential buildings

In this practice exercise you should try various approaches to searching while directing your efforts to the goals specified above. In the future, determine before beginning to search whether you are looking for a few relevant items on the subject or whether you are after an exhaustive bibliography.

Take a moment to think of some terms which are likely to appear in citations on the topic you have chosen. Note terms which you will use in developing your search. For example, you might plan to EXPAND the term 'solar' to see terms useful in a search on 'solar heating'. Or consider EXPANDing the term 'asphalt' for terms which may be useful in a search on 'asphalt paving.'

To assist you in your search, we would like to know the approximate size of the final set you intend to create. Please enter a number to indicate this maximum size.

I? 30

IIDA will now place a telephone call to the DIALOG retrieval system via the TYMNET telecommunications network. This may take a minute or two.

EXHIBIT 5 -- DIALOG Logon for Assistance Mode

```
Enter your IIDA identifier:  0099021pa

          Welcome to IIDA.  Would you like a brief introduction to
the use of the system?  (y or n)

I? n
To begin one of the following enter its number:
          Exercise 1: basic DIALOG commands
          Exercise 2: practice DIALOG search
          Exercise 3: advanced commands and search strategy
                    4: search assistance only
Enter '/done' if you are finished with IIDA.

I? 4
          To assist you in your search, we would like to know the
approximate size of the final set you intend to create.  Please
enter a number to indicate this maximum size.

I? 20
          IIDA will now place a telephone call to the DIALOG
retrieval system via the TYMNET telecommunications network.  This may
take a minute or two.
Telephone 1 will be used.
11/20/79  1624.1 est Tue
delay:                          355843     *1*b
11/20/79  1624.5 est Tue
Phone connection made to TYMNET.
From host: nti
TYMNET responding
Logging onto DIALOG.
News from DIALOG:
WWWWWWWW  LOGON File1 Tue 20nov79 16:25:32 Port04E
FILES 3 AND 16 ARE OFFLINE . . .
 DIALOG AVAIL. REGULAR HOURS DURING U.S.
 THANKSGIVING HOLIDAY PERIOD.
REGISTER ONLINE FOR BEST OF UPDATE IN
LONDON ON DECEMBER 7.  SEE ?NEWS.
NEW DISCLOSURE (FILE 100) NOW AVAILABLE
 WITH HALF HOUR FREE CONNECT TIME IN NOV
WHITE HOUSE CONF. HOTLINE (FILE 202).
You are now connected to Lockheed DIALOG retrieval system--
Choose data base 1, 6, 8, or 201 with the BEGIN command

D? begin1
          20nov79 16:25:59 User7303
  $0.14  0.009 Hrs File1*
File1*:ERIC 66-79/OCT
          Set Items Description (+=OR;*=AND;-=NOT)
          --- ----- ---------------------------
```

EXHIBIT 6 -- Failure to log onto DIALOG

```
Telephone 2 will be used.
11/21/79  1154.8 est Wed

delay:                              335240     '0'b
11/21/79  1156.2 est Wed
call: Unable to complete connection to external device.
unable to phone the TYMNET network.
          Since our call via TYMNET was unsuccessful, IIDA will try to
place a call to DIALOG via the TELENET telecommunications network.
Again, this may take a minute or two.
Telephone 2 will be used.
11/21/79  1156.4 est Wed
delay:                               85645     '0'b
11/21/79  1157.6 est Wed
call: Unable to complete connection to external device.
unable to phone the TELENET Network.
The login process has failed.
To quit IIDA, enter: stop.
To try phoning Telenet, enter: ph te
To try phoning Tymnet, enter: ph ty

I?


   stop
```

EXHIBIT 7 -- Exercise 3: Frames 1-8

Exercise 3 introduces some new commands and techniques and reviews some of the material you learned in Exercise 1. You will see eight frames which describe the contents of the exercise, after which you will have two options:

(1) You may use the eight frames as a table of contents for the help library when you go on to do an assisted search. You may access the detailed information when you use the help library.

(2) You may use the eight frames as an introduction to Exercise 3 and when you have seen all eight, go back and take the detailed instruction. You will be instructed on how to do the units after you've seen the eight frames. You will have the opportunity to practice the new commands and techniques if you choose this option.

Remember that you may correct errors by entering @ to erase a line and # to erase one or more characters. Then enter the correct command and follow your entry with a carriage return.

Contents of Instructional Units

a (1) Review of Basic Commands
- begin, expand, select, page, combine, type, logoff

b (2) Advanced Commands
- file selection, limiting, type and print, abbreviati
time-saving techniques, search save

c (3) Text Searching
- searching for words or phrases embedded in larger sentences or paragraphs

d (4) Search Strategy
- planning the search in advance

e (5) Database Descriptions
- contents and use of data bases

f (6) Beginning and Ending the Search
- how to initiate your conversation with DIALOG and how to sign off when you're finished

g (7) IIDA Facilities
- how to use IIDA during your search

Strike the carriage return when you are ready to see the next frame.

Basic Commands

In order for you to perform a search an understanding of six basic commands is necessary. The commands are:

a (11) BEGINn - to start a new search and select a file
b (12) EXPAND - to view the data base dictionary
c (13) SELECT - to create a set
d (14) COMBINE - to combine two or more sets
e (15) TYPE - to display records at your terminal
f (16) PAGE - to browse through the dictionary

Definitions for the basic commands are available for review.
Strike the carriage return when you are ready to see the next frame.

Advanced Commands
Use of a few commands and techniques beyond the basic commands will enable you to save time and perform your search more efficiently.
Explanations for the following advanced commands and techniques are available:

a (21) Abbreviations of commands
b (22) Short cuts for command entry
c (23) DISPLAY SETS command
- to review previously created sets: DS
d (24) Limiting
- how to reduce the size of a set with subject terms
e (25) TYPE and PRINT
- how to display records
f (26) File Selection
- how to choose a new file to search: .FILE, BEGINn
g (27) Saving the Search for later execution

Strike the carriage return when you are ready to see the next frame.

Text Searching
Text searching enables you to search for words or phrases in the text of a title or abstract. It is distinguished from a term search in that in text searching you do not specify the entire content of an information element, just a word or phrase within it. For example, to search on an author's name using a term search, you must give the name exactly as it is stored in the computer. To find any occurence of the term 'shale oil', you can ask the computer to look for this in any part of a title or abstract
Instructional units are available on the following text searching techniques

a (31) Truncation - searching on an incomplete term or word stem
b (32) Prefixes & Suffixes - searching of particular parts of the database
c (33) Infix operators - specifying word proximity and order

Strike the carriage return when you are ready to see the next frame.

Search Strategy
To make the best use of on-line facilities, a preplanned approach to the search is essential. This plan is called a Search Strategy.
Details are available on:

a (41) Planning the Search
b (42) What to do about null sets
c (43) Logical search formulations
d (44) Sequences of commands

Strike the carriage return when you are ready to see the next frame.

Databases
Descriptions of four databases are available for your review. Each description includes information about the subject of items in the file, sources of information, and particular instructions for searching the file. Database-dependent techniques such as LIMITing, using prefixes and suffixes, and using various TYPE formats are spelled out in detail.

a (51) ERIC File 1 - Educational Materials
b (52) NTIS File 6 - Government-sponsored research and reports
c (53) COMPENDEX File 8 - Engineering Index
d (54) ONTAP ERIC File 201 - Online Training & Practice (Education)

Strike the carriage return when you are ready to see the next frame.

Beginning and Ending a Search
The instructions spelled out in this unit are available from the IIDA assistant.
The following instructions are available on-line for review:

a (61) Getting on the System
- how to connect your terminal to the computer and the retrieval system

b (62) Beginning the Search
- how to initiate your conversation with DIALOG and picking a database

c (63) Ending the Search
- how to sign off when you complete your search

Strike the carriage return when you are ready to see the next frame.

IIDA Facilities
IIDA is designed to help you with actual searches as well as providing basic instruction in search commands and techniques. IIDA can help you in three ways:

1. IIDA is flexible. You may ask IIDA for quick advice by entering /aa . Assistance with choosing descriptors is available by entering /da (descriptor assistance). Finally, if you feel that IIDA is raising too many conditions and sending you too many messages, enter /slack . IIDA will then slacken off.
2. IIDA diagnostic facilities will be with you throughout your search to point out unproductive conditions and offer remedies. You have the choice of accepting IIDA's recommendations or of continuing as you were. IIDA will merely remind you of such occurrances as null sets, unused sets, and similar conditions which may affect your results.
   a (71) Diagnostics
3. IIDA's /HELP facilities enable you to ask for help when you feel you need it. Besides /help for the table of types of help, you may enter any of the following abbreviations for help during the search:
   - /h1 - definition of commands (the library)
   - /h2 - advice on current problems
   - /h3 - commands given in this search
   - /h4 - sets created in this search
   - /h5 - summary of records viewed in this search
   - /h6 - summary of errors made
   - /h7 - summary of descriptors used
   - /h8 - change to another exercise or mode

Strike the carriage return when you are ready to see the next frame.

You have seen the introductory frames of Exercise 3.
You now have two options:
(1) to see more detail on any of the topics listed above, enter the two-digit frame number preceding the topic of interest.
(2) To begin your search in Assistance Mode, enter '/done'.

Enter a letter, any frame number, or /done
/done
Returning to the main program.

EXHIBIT 8 - Detailed Instructional Frames - Exercise 3

Special characters are found in the first column on many lines. They will be explained in context in Report #6; a summary interpretation follows:

Character	Explanation
%	A 5-digit frame number follows
*	Pause for user response (not in Exercise 3 as HELP)
^	User response must match one of these lines (not in Ex. 3 as HELP)
$Enter...	Print this line only when frame is accessed as HELP
$(all else)	Print only these lines when frame is accessed as a menu in HELP
~	Print the present time and date

```
%00011
BEGINn
The BEGINn command is used to start a search.  The most direct way
to start the search is to use BEGIN with n, the number of the file
you want to search. To do this, enter
BEGINn
where n is the number of the database you want to search.
There are four files available to you through IIDA:
File 1 (education), File 6 (government sponsored research,
technical reports) , File 8 (engineering) and File 201
(abbreviated ERIC file, 1975 records only).
The system response to your BEGIN command is date, time,
and a serial number along with cost data and time in the last
file.  The file number and name of the file you have selected
will also be returned.  When you receive a prompt, enter
your next command.  When you use the BEGIN command all previous
set numbering is restarted at one, work done to that point
is erased, and timing starts anew.
*Enter BEGIN1
^begin1
$Entering BEGIN1, for example, yields:
~ User7005
  $0.08  0.005 Hrs File1
File1: ERIC
       Set Items Description (+=OR;*=AND;-=NOT)
        --- ----- ----------- -----------------
You may abbreviate the BEGIN command by entering B and the file
number. Information is available on use of the full BEGIN, which
is used by searchers who perform numerous searches each day,
and on use of BEGIN to change files when searching several files
at the same online session.
$More detail is available on:
$          a (111) Full BEGIN
$          b (112) Changing Files
$Enter a DIALOG command or the letter of the unit you wish to see.
%00012
EXPAND
The EXPAND command enables you to examine an index of terms related
either alphabetically or by subject to the term which is of interest
to you. Expanding a term to be used in the search does not create
sets of citations but helps you to identify useful terms which
you will use later in set-creating steps of the search. To examine
terms related to the term which is of interest to you, enter
EXPAND term where term is the word or phrase of interest.
*Enter EXPAND INFORMATION RETRIEVAL
^expand information retrieval
^e information retrieval
^einformation retrieval
```

```
$Entering EXPAND INFORMATION RETRIEVAL, for example, yields:
Ref   Index-term                Type Items RT
E1    INFORMATION PROCESSING
        MODEL----------------       1
E2    INFORMATION PROCESSING
        THEORY---------------       2
E3    INFORMATION PROCESSOR
        MODELS---------------       1
E4    INFORMATION PRODUCTION--      1
E5    INFORMATION REFERRAL
        CENTERS--------------       2
E6   -INFORMATION RETRIEVAL---    2191 22
E7    INFORMATION RETRIEVAL
        (PSYCHOLOGICAL)-------      2
E8    INFORMATION RETRIEVAL
        CENTER ON THE DISADV--      2
E9    INFORMATION RETRIEVAL
        DEMONSTRATION AND RE--      1
E10   INFORMATION RETRIEVAL
        PRECISION------------          1
E11   INFORMATION RETRIEVAL
        TELEVISION-----------       3
                                  -more-

The column headed 'Ref' lists E-reference numbers for the term;
under 'Index-term' are the terms related alphabetically to your term;
under 'Items' are the number of items in the database which use the
terms.  You may abbreviate the EXPAND command by simply entering
E instead of the word EXPAND.

$More detail is available on
$          a (121) Expanding authors names
$Enter a DIALOG command or the letter of the entry you wish to see.
X00013
Basic Commands - SELECT
SELECT is the first set-creating command used for a search.  SELECTing a
term creates a set of records containing that index term.  The term
may be a single word or a multiple-word phrase.  A set created by the
SELECT command will most likely be combined with other sets by use of the
COMBINE  command.  You may SELECT from a previous EXPAND table or you
may SELECT a term directly.  The system will respond with a unique set
number, the number of items in the set, and a recapitulation of the term or
terms selected.  To use the SELECT command, enter SELECT term where
'term' is the term of interest, or SELECT En where 'n' is the E-reference
number from the previous EXPAND list.
*Enter SELECT COMPUTER
~select computer
$Entering SELECT COMPUTER, for example, yields:
          1 11467 COMPUTER
The SELECT you have just entered is the direct SELECTion of a term.  A direct
SELECT requires that the spelling of the term as you enter it equal the
spelling of the term as the term is indexed; otherwise you will receive a null
set (no postings for the term entered).  If this happens, it might be wise to
EXPAND the term or try another spelling.
To SELECT from an EXPAND table, enter SELECT and  the E-reference number.
*Enter EXPAND COMPUTER
~expand computer
$Entering EXPAND COMPUTER, for example, yields:
Ref   Index-term                Type Items RT
E1    COMPUTED---------------     575
```

```
E2    COMPUTED ASSISTED
        INSTRUCTION----------        1
E3    COMPUTEER------------        1
E4    COMPUTEK-------------        1
E5    COMPUTEK MODEL 400-20
        COMPUTER-------------        1
E6   -COMPUTER-------------- 11467
E7    COMPUTER ACCESSED
        MICROFICHE LIBRARY----       1
E8    COMPUTER AIDED DESIGN---       3
E9    COMPUTER AIDED DIAGNOSIS       1
E10   COMPUTER AIDED
        INSTRUCTION----------            1
E11   COMPUTER AIDED
        INSTRUCTION CENTER----       1
E12   COMPUTER AIDED LEARNING-       5
E13   COMPUTER AIDED LEARNING
        PACKAGES-------------        1
                                   -more-

*Enter SELECT E2
^select e2
$Entering SELECT E2, for example, yields:
          2     1 COMPUTER ASSISTED INSTRUCTION
                      E6: COMPUTER
You may abbreviate SELECT by entering S term or S En instead of spelling
out SELECT.  It is possible to SELECT a range of terms from an EXPAND table
and to SELECT many related terms based on the same term, even if you are
unsure of the exact term as indexed.
$More detail is available on
$          a (131) SELECTing a range of terms
$          b (132) SELECTing when unsure of exact term (truncation)
$Enter a DIALOG command or the letter of the entry you wish to see
%00014
Basic Commands - COMBINE
The COMBINE ccmmand is used with logical operators AND, OR, and NOT
to create new sets from combinations of other sets or by
excluding elements from already created sets.  The COMBINE command
is always used with set numbers, not with terms or groups of terms.
To use the COMBINE command enter COMBINE x logical operator y
where x is one set number and y the other set number and logical operator
is either AND, OR, or NOT.
*Enter S COMPUTER: S INSTRUCTION
^S COMPUTER: S INSTRUCTION
^s computer: s instruction
^s computer:s instruction
$Entering S COMPUTER: S INSTRUCTION, for example, yields:
          3 11467 COMPUTER
          4 46721 INSTRUCTION
*Enter COMBINE 3 AND 4
^combine 3 and 4
^combine 3and4
^c 3 and 4
^c 3and4
^c3and4
$Entering COMBINE 3 AND 4 yields:
          5   147 3 AND 4
The response to your COMBINE command is a new set number,
the number of items in the set, and a recapitulation of the sets
combined.
```

Since you used the logical operator AND, the resultant
set consists of those items in the file which use both terms.
You may abbreviate the COMBINE command by entering C instead
of COMBINE.

$More detail on the COMBINE command is available.
$ a (141) Use of Logical Operators
$ b (142) Combined Logical Operations
$Enter a DIALOG command or the letter of the entry you wish to see.
%00015
Basic Commands - TYPE
The TYPE command is used when you want to see items from a particular
previously retrieved set, when you have completed your search, or
when you want to browse through a number of items to determine
if you are near your search goal. Use the TYPE command by entering
TYPE s/f/m-n where s is the set number, f is the number of the bibliographic
format in which you wish to see the item, and m - n are the numbers, inclusive
of the items you wish to see.
Eight formats are available for most of the database files. The formats
which you will use most often are formats 2 (gives the bibliographic
record except for the abstract), 6 (gives title only), and 5 (gives the
bibliographic information and the abstract as well). For example:
*Enter TYPE 10/6/1
~ type 10/6/1
~type 10/6/1
~t 10/6/1
~t10/6/1
10/6/1
$Entering TYPE 10/6/1, for example, yields:
ED121290#
Impact of On-Line Retrieval Services: A Survey of Users, 1974-75.

The number displayed is the accession number of the document. Format
6 is helpful in some cases, but you will probably find formats 2 and
5 give you more needed information.
If you fail to specify the number or range of items to be typed,
the system will automatically give you the first item (most recent
addition to the file). If you fail to specify a format, the system
will default to Format 2 and give you the
first citation. It is preferable to specify the set number, format,
and number of items you wish to view to be certain you get all the
information you need. The TYPE command may be abbreviated simply T.
$Details on the eight TYPE formats
$and on the direct TYPE are available:
$ a (151) TYPE formats
$ b (152) Direct TYPE with Accession Number
$Enter a DIALOG command or the letter of the details you wish to see.
%00016
Basic Commands - PAGE
The PAGE command is used to continue the display of a table or
record. PAGE is used with the EXPAND command when the number
of related items to your term exceeds the number which can be
displayed in one frame.
The word "-more-" appearing beneath a table or list denotes
that the listing is incomplete. To view additional lines,
simply enter PAGE as many times as needed to complete the listing.
If all the information you need is displayed on the first
page, then disregard the "-more-" and continue with your search.
%00021

Abbreviations
Each keystroke taken to enter a command takes time and costs money. Therefore it is good practice to abbreviate the command and close up the space between the command and its argument. The following table gives commands, abbreviations, and examples.

COMMAND	ABBREVIATION	EXAMPLE OF USE
BEGINn	Bn	BEGIN1 = B1
EXPAND	E	EXPAND unions = Eunions
SELECT	S	SELECT unions = Sunions
COMBINE	C	COMBINE 1 AND 2 = C1AND2
TYPE	T	TYPE 5/6/1-2 = T5/6/1-2
LIMIT	L	LIMIT 1/MAJ = L1/MAJ
LIMITALL	LALL	LIMITALL/MAJ = LALL/MAJ
LIMITALL/ALL	LALL/ALL	LIMITALL/ALL = LALL/ALL
PAGE	P	PAGE = P

You may also abbreviate the logical operators:
AND = *
OR = +
NOT = -
Recall that it does not matter whether a space is entered between the command and its argument, and that upper or lower case may be used to enter commands. Closing up the spacing and using lower case will save you time.
%00022
Short-cuts
Just as abbreviating commands will save time, so will the use of two basic short cuts. Stacking consists of entering several commands on one line, separated by semicolons (;), followed by a single carriage return. The set range short cut, which allows you to COMBINE many sets, using the same logical operator, will save you the trouble of entering a long succession of COMBINE commands.
$Details on the use of two short cuts are available:
$ a (221) Stacking
$ b (222) Set range short cut
$Enter a DIALOG command or the letter of the entry you wish to see.
%00023
Display Sets Command
The DISPLAY SETS command may be used at any time you receive the prompt. This command is used when you wish to see a review of the sets you have created thus far in your search. The system will respond with a listing of all the sets created.
This reponse will allow you to see, at a glance, what terms you have selected and which sets you have combined with other sets by the use of logical operators. The DISPLAY SETS command may be abbreviated simply "ds".
%00024
Limiting
Two advanced commands, LIMIT and LIMITALL, can be used to trim the search output. The limiting commands restrict the sets already created according to particular criteria such as publication date or a specific range of accession numbers. LIMIT differs from other commands you have used in that it is dependent on the particular database in which you are working. LIMIT applies to previously created sets; LIMITALL is a condition you specify which you want to apply to sets that will be created in the future.
Use LIMIT by entering LIMIT s/suffix, where s is the set number and suffix is the particular suffix code for the criteria by which you want the set or sets restricted.
*Enter SELECT COMPUTERS

```
~select computers
$Entering SELECT COMPUTERS, for example, yields:
          8     99 computers
*Enter LIMIT 8/MAJ
~limit 8/maj
$Entering LIMIT 8/MAJ yields:
          9     23 8/MAJ
You have limited the set created by the SELECT command to those
citations for which COMPUTERS is used as the MAJOR descriptor,
or the primary term which  describes the contents of the
document. Note that by limiting by MAJOR descriptor you have
trimmed the size of the set considerably.
You may abbreviate the LIMIT command by entering simply L with the
appropriate suffix. The other criteria by which you can LIMIT are
discussed in more detail in instructional unit 241.
The LIMITALL function performs similarly, except that upon
entering LIMITALL you will limit all subsequently formed sets. Hence,
the use of LIMITALL does not require a set number since the LIMITALL
condition applied to sets that will be created. To use the command,
enter LIMITALL/suffix where "suffix" is the appropriate suffix which
specifies a limiting restriction.  The same suffixes are used with
LIMIT and LIMITALL.
When you wish to cancel the LIMITALL condition, enter LIMITALL/ALL
and future sets will not be limited.
$More detail is available on LIMITING suffixes:
$          a (241) Limiting criteria and suffixes
$Enter a DIALOG command or "a" if you wish to see the detailed information.
%00025
TYPE and PRINT
The TYPE command is used when you want to view items from a particular
previously retrieved set. To use the command enter TYPE s/f/m-n where
s is the number of the set of interest, f is the format number for
the format in which you wish to see the items displayed, and m-n is
the range of items you wish to see Format options are:
          1 - Accession number
          2 - Full record except abstract
          3 - Bibliographic citation
          4 - Abstract and accession number only
          5 - Full record
          6 - Title only
          7 - Bibliographic citation without indexing
          8 - Title with subject indexing only
The PRINT command is used with the same format numbers, and the command
is used in the same way.  Instead of printing the items on-line,
the citations are printed off-line and mailed to you. The PRINT
command is not available for your use at the present time through
IIDA. It is a good service to take advantage of when you perform
searches in the future: it will save on-line time and cost. When
you do use the PRINT command, it is still good practice to TYPE
out a few of the items from your final set before PRINTing to
be certain that the items meet your search requirements.
%00026
File Selection
BEGINn and .FILEn are commands used to begin searching in a particular
file.  Either command may be used to change files during the course
of your search.  The quickest way to begin any search is to use
the command BEGINn. (n denotes the number of the file you wish to
search in.)  This command is abbreviated simply Bn.
Use of the BEGIN command causes all work done up to that time to
be erased and set numbering to begin again at 1. Time and cost
```

data is reported whenever you use BEGINn. Each time you plan to change files or begin searching a new topic altogether by using BEGINn, you should complete all operations and displays of items in the "old" file unless you want to recreate all the sets or use the SEARCH SAVE feature of the system to save your search.
A second file selection command is .FILEn.
It is used when you want to change from one file to another, but want set numbers to continue without beginning again at 1. When .FILEn is being used, the system gives the same response as when BEGINn is used, i.e., time and cost data for the previous file are returned.It is best, when doing several searches in succession, to use the BEGINn command as repeated use of .FILEn may cause your searches to exceed the system's capacity of 98 sets.

```
*Enter BEGIN1
~begin1
$Entering BEGIN1, for example, yields:
~ User7005
  $0.09  0.006 Hrs File1*
File1*:ERIC 66-79/FEB
         Set Items Description (+=OR;*=AND;-=NOT)
         --- ----- ---------------------------

*Enter .FILE8
~.file8
$Entering .FILE8 yields:
~ User7005
  $1.11  0.074 Hrs File1* 2 Descriptors
File8:COMPENDEX 70-79/APR
(Copr. Engineering Index Inc.)

The details of executing a search in several files are explained
in instructional unit 261.
$More detail on searching in multiple files is available.
$          a (261) Using Multiple Files
$Enter a DIALOG command or the letter 'a' to see detailed instructions.
%00027
Saving the Search Formulation
The system has a SEARCH SAVE feature which allows you to save your search
formulation for future instances when you might want to perform
the same search, using the same commands and arguments, in another
file, on a different class of items, or on a more recently acquired
set of documents in the database file. The saved search formulation
will save you the time necessary to re-enter all the commands and
the on-line time spent waiting for responses to be returned before
entering subsequent commands. SEARCH SAVE stores your strategy until
you want to execute the search; it does not store the postings
associated with the original search.
When you are searching with IIDA's assistance mode you may not
use the SEARCH SAVE feature, but you may review details of
the use of the feature for future reference.
$Instructions are available on:
$          a (271) Use of SEARCH SAVE commands
$Enter a DIALOG command or the letter 'a' to see the detailed directions.
%00031
Truncation
The truncation or variable character symbol "?" is used to search on a
word stem or part of a word or term. The system will automatically SELECT
up to 800 terms based on the stem you specify and create a set of all the
items using that stem. The truncation technique is useful when you need to
SELECT both singular and plural forms of a word or when you are uncertain
```

as to the form a term takes in the index. Truncation may be used at the end of a word, or the variable character may be embedded within a word. For example, SELECT LIBRAR? will yield items such as library, libraries, librarians, and librarianship as index terms. Similarly, SELECT NAVA?O will create a set with items which use the terms NAVAHO as well as NAVAJO. A third example of how truncation may be used permits SELECTion of a term with a specific number of variable characters following the stem. For example, SELECT librar??? ? will limit the SELECT to terms with the LIBRAR stem and three characters following the stem.

To use truncation with a specific number of variable characters, enter the SELECT command, question marks to represent however many variable characters you want to follow the word stem, a space, and finally one question mark. This is one instance in searching where you should not close up the spacing between groups of characters, so please do not omit the space.

```
*Enter s wom?n
~ s wom?n
~s wom?n
~swom?n
$Entering S WOM?N, for example, yields:
          1  7158 WOM?N
*Enter s woman
~ s woman
~s woman
~swoman
$Entering S WOMAN yields:
          2   667 WOMAN
```

Note that set number 1 is considerably larger than set number 2 as a result of using truncation to collect all entries for woman and women.

Several cautions regarding truncation: it is best to truncate at the latest possible point within a word. Otherwise many irrelevant terms will be included in the set created. For example, SELECTing COM? rather than COMPUT? to get terms based on computer and computers would retrieve compact, comprehensive, commander, common, and many other terms not at all related to your search.
Secondly, you may receive a system message that over 800 terms are based on the stem you SELECTed. If this occurs, attempt to use SELECT with a longer stem, or respecify the stem by entering several terms with longer stems.
%00032
Prefixes and Suffixes
It is frequently helpful to be able to limit searching to terms that are found in particular fields of the citations. Because the files are comprised of different fields, such as author, corporate source, subject indexing categories, it is possible to search these subfiles instead of searching the entire basic index for each term or concept of interest. Prefix codes are used to search file terms which are not part of the basic index; suffix codes are attached to terms to designate that searching is to be limited to specific subfiles or parts of the basic index. The basic index, which is the main subject-conveying word file for the database, is searched if no suffix has been designated.
Prefix and suffix codes are used with the EXPAND and SELECT commands. To use prefix codes for searching particular subfiles, syntax requires that you enter the two-character prefix, an equal sign (=), and the term. For example, to EXPAND an author's name to insure that you have found all possible variations and initializations, enter

EXPAND AU=name, where "name" is the author's last name.
The system will in turn EXPAND only the author entries alphabetically related to the name which is of interest to you. Using the AU= prefix is one of the most helpful techniques to use when searching for works by particular authors and researchers.
Other prefix coded fields include journal announcement dates, journal names, and others depending on the individual databases.
To use the suffix codes, enter the SELECT or EXPAND command, the term of interest, a slash mark (/), and the suffix code for the subfield you wish to search in. For example, entering SELECT ENERGY/DE limits the search for items with "ENERGY" used as a descriptor, and does not search the other fields within the basic index. Similarly, entering SELECT ENERGY/TI would find items where "energy" appears in the title field. The fields which can be identified by suffix codes include: title (/TI), descriptor (/DE), full descriptor (/DF), identifier (/ID), descriptors as primary terms (/DF*, /DE*). Other searchable fields depend upon which database you are searching. Multiple suffixes may be used, but the suffixes must be separated by commas. By increasing the number of fields for searching you will increase the likelihood of finding items. The suffix codes cannot be used with truncated terms: separate SELECT statements must be made for each variation of the term you SELECT. A null set will result from using suffix codes with truncated terms.
$Details are available on:
$ a (321) Using Prefix Codes
$ b (322) Using Suffix Codes
$ c (323) Using Multiple Suffix Codes
$Enter a DIALOG command or the letter of the entry you wish to see.
?00033
Full Text Searching - Infix Notations
The retrieval of citations by use of multiple-word search terms is possible because of the system's capacity for full-text searching. Full text searching allows you to specify multiple word phrases as your search term, when the multiple-word phrases are not descriptors, but appear in titles, abstracts, and other fields.
Full text searchable fields are identified in the individual databases by suffix codes, and include title, descriptor, identifier, in all IIDA files: others such as abstract, corporate source, and sponsoring agency are database-dependent. It is possible to do full text searching in the basic index or in specified suffix-coded fields. Infix notations, which are symbols that set order and word proximity within the multiple word phrase, permit you to be as specific or as general as you wish.

Infix notations - word proximity operators
Word proximity operators are symbols entered between the words in a multiple-word term to denote ordering, location, and proximity of terms within the searchable fields. By using the infixes you will have control of where the words appear in the citation and where they appear in relation to other words within the multiple-word term. You may, for example, find citations on "law school" by entering SELECT law, SELECT school, and by combining the two terms with an AND COMBINE command. However, this would give you many irrelevant items as you would also retrieve items on "school law." Therefore, it is more efficient to designate that you want to search for the two terms - appearing in the same field, in specific order , and with no intervening words. To do this

you would enter SELECT LAW(W)SCHOOL; the (W) is the infix notation which restricts the search to items where "law" and "school" are directly adjacent, in any suffix coded field, and in the order specified (that is, "law" appearing first, followed by "school").
There are five infixes which are used in the IIDA available files.
$The infixes and their use are explained in more detail:
$ a (331) Infixes and Their Use
$Enter a DIALOG command or the letter "a" to see the detailed information.
%00041

Planning the Search

A written list of terms and descriptors, with alternative terms to search for, is helpful in saving time and preventing duplication of commands and terms. Determine in advance the major concepts of your search topic. The following steps will be of assistance:

(1) Define your search topic in as much detail as possible.
(2) Break the search topic into facets of individual concepts.
(3) Select some terms that may be used as descriptors before going on-line. The use of a printed thesaurus may eliminate costly EXPAND commands.
(4) Use the TYPE command to browse when you are near your search goal.
(5) Follow a logical search formulation.

%00042
Null Sets
A SELECT command or an AND or NOT COMBINE command may yield a NULL SET - a set which has no contents - that is, no citations at all. When you receive a null set, the number of postings given after the set number will be -0- . By using a null set in future combinations of search formulation you will achieve nothing; therefore you might try EXPANDing the term selected or if the null set is the result of a COMBINE command you might attempt another approach to restricting or narrowing the search. An alternative approach, if you are searching on a very specific term or multi-word term is to try SELECTing a broader term.
If you SELECT terms restricted by suffixes (either LIMITing or subfield suffixes used alone)and get a null set, you might try searching in more subfields or make the LIMITing criteria less restrictive. If you produce null sets by using the full text operator (w) you might try allowing more terms between the terms connected with the infix. You can even try SELECTing each term separately, and then using the COMBINE command to put the terms together.
Keep in mind that throughout your search you will be making decisions as to which approach to take. Search formulation is, by its nature, dynamic, but there is great value in planning, in advance, at least the approach you will be taking. Planning your strategy in advance will save you on-line time and make your total search success more certain.
%00043
Logical Strategies
To formulate a logical approach to your search, first define the search topic in some detail. Identify facets or concepts first and proceed from there. The more synonymous terms you search on the more opportunities there will be for a match to occur between the terms of interest to you and the terms used in the file.
Recall that the AND and NOT logical operators narrow the search while the OR logical operator broadens the search and causes

more output. You will use the logical operators according to how specific or general you want your search to be. It is useful to plan not only the topic or subject of the search, but also to predetermine the number of citations you need, and then to set a search goal which corresponds with your need for information. There are a number of different approaches you may take to the search.

1. The MODULAR APPROACH first treats each facet of the search as if it were a single search topic and then assembles the subparts. For example, to find citations on software acquisition management, you would develop the terms "software", "acquisition", and "management" first, and then you would combine them using the "and" logical operator.
   To use the MODULAR APPROACH, use as many "or" COMBINE statements as needed and then use "and" with COMBINE to put all the parts of the search together.
2. The SPECIFIC TO GENERAL approach first searches on the most specific term in the query and then broadens the search from there. This approach takes more time than the modular approach but can yield very particular results.
   To use this approach, first search on the most specific term in the topic and find and type out one citation. Type the citation in Format 2 so that you will get the list of descriptors. From the list of descriptors, build the search by looking for new terms and then combine those with terms derived from looking at citations from the other facets of your topic.
3. A third approach is to use the FILE PARTITIONING approach. This approach involves searching on very broad terms, viewing some citations that result from a very general search, and then cutting down the size of the search by using the NOT logical operator with terms which do not pertain to the subject of interest. For example, if you wanted information on "library schools" you might first search on such terms as "library" and its variants, "schools", "library education" and others. Then you would build a set which combines the very general facets. Then, by looking at the descriptors (by using format 2 or 8) you could combine the final set with terms in which you are not interested: you might use "not" logical operator, in this case, with such descriptors as "continuing education", "inservice training", and others. This will decrease set size.
   Using this form of strategy you may also use the LIMIT command and other techniques which are reviewed in the instructional unit on advanced commands.

X00044
Sequence of Commands
The sequence which is generally followed in developing a search formulation is as follows: EXPAND, SELECT, COMBINE, and TYPE. This sequence of commands represents a cycle. Each time you build a new facet of the search you will probably go through another cycle. When you have developed a cycle from expanding and selecting terms, and combining the facets of the search with other facets it is wise to type out a few citations to "browse" through the search results. In this way you will know if your search results are relevant. If so, then you would type out the citations in the longer format, but if the results are not to your liking you can reformulate a part of your strategy before printing out many citations in the longer formats.
By reformulating facets of your search you will have many opportunities to reach your search goal.

X00051
ERIC File 1

ERIC is the complete database on educational materials from the Educational Resources Information Center. Subjects covered include career education, school counselling and personnel services, education from early childhood to higher education, teaching and educational materials on subjects within the science, social sciences, and humanities, fields, teaching skills, information resources, and educational programs for special groups such as handicapped , urban, and rural groups.
ERIC consists of two main files: Resources in Education, which identifies education research reports, and Current Index to Journals in Education, an index of over 700 publications in all areas of education. The ERIC file contains research reports, lesson plans, bibliographies, journal articles, course descriptions, evaluation studies, and even pamphlet materials. Except for journal articles, most of the materials can be procured through ERIC.
The ERIC file includes materials from 1966 to the present, is updated monthly with about 3000 items, and contains approximately 300,000 citations. The ERIC is produced by the Education Resources Information Center, National Institute of Education, Washington, D.C. 20208.
$ More detail is available on search methods and formats in ERIC.
$ a (511) ERIC - Search Methods & Formats
$ b (512) Expanding related terms (RT)
$ c (523) Expanding directly to subject-related terms
$Enter a DIALOG command or the letter of the frame you wish to see.
X00052
NTIS - File 6
NTIS is the database which includes government-sponsored research, development, and engineering reports covering a large variety of social science and science fields. NTIS stands for "National Technical Information Service."

NTIS includes analyses prepared by federal agencies and their contractors. Reports of NASA, DOE, HEW, HUD, DOT, the Department of Commerce, and other agencies are available through NTIS. Subjects covered include the hard and soft sciences with substantial coverage of technological applications, business procedures, and regulatory matters. NTIS is the means by which unclassified, unlimited distribution reports are made available to the general public.
Included are reports from 1964 to the present; updates are bi-weekly and about 5000 items are added monthly.
$More detail is available on
$ a (521) Search methods and formats in NTIS
$Enter a DIALOG command or "a" to see the detailed information.
X00053
COMPENDEX - File 8
COMPENDEX is the machine readable version of the Engineering Index. It provides engineering information abstracted from the world literature of engineering and related disciplines. It covers journal, society, and organization literature, as well as proceedings of conferences and selected books and reports.
COMPENDEX covers all areas of engineering, including civil, environmental, electrical, electronic, aerospace, geological, fuel, and other areas. Industrial and management applications are also included.
Publications indexed include 1800 journals, and 1000 works from conferences, symposia, etc. Items in the file date from 1970 to the present; the file is updated monthly with about 7000 citations a month.
$More information is available on
$ a (531) Search methods and formats
$Enter a DIALOG command or "a" to see the additional detail.

%00054
Ontap Eric
ONTAP ERIC is a specially programmed training file which consists of the 1975 ERIC databases as the source of citations. The file (File 201) is used similarly to the ERIC database (file 1). If you choose to search in ONTAP ERIC File 201, the set sizes will be much smaller than if you search in ERIC because of the fact that ONTAP ERIC covers only one year of the database content. It is also necessary to note that certain commands, especially the Search-Save commands, cannot be used when searching in Ontap Eric.
For more detailed information on use of ERIC generally, return to the instructional unit on ERIC.
%00061
Getting on the System
You are connected to the on-line system by telephone. Therefore, the first step in getting on the system is to dial the number of the network communications system.
Procure the number from your IIDA assistant and dial.
Upon hearing the steady high-pitched tone, follow the steps outlined below.

1. Turn on the terminal and turn on the acoustic coupler.
2. Place the telephone receiver into the acoustic coupler. A light should appear on the coupler when the phone handset is properly inserted. (A symbol or the word "cord" should appear on the coupler to indicate which way to place the handset.)
3. Enter two carriage returns.
4. The system will respond with the message "terminal=". Enter one carriage return.
5. You will receive a prompt, an "a". Then enter "c 617 ms" and one carriage return.
6. You will receive a message which looks like the following:

```
Multics 34.3: MIT, Cambridge, Mass.
Load = 45 out of 85.0 units:  users = 45
```

7. When this message has been received, you should enter the "login" message and name that you have been assigned. After you have entered the "login" message, enter one carriage return.
8. The system response to no.7 is a request for a PASSWORD. The system will print over the request for the password. When the typing stops, enter the four character password which has been assigned. You will not see the notation, as you will have printed over other characters. This is designed to maintain security. Enter one carriage return after entering the password.
9. You will receive a message and time, date, and last "login" time and date. IIDA immediately begins at this point.

%00062
Beginning the Search
After getting on the system you will begin your search with the command BEGINn where n is the number of the file you wish to search. You may abbreviate BEGIN by entering simply B with the number of the appropriate file. If, during one on-line session, you wish to change files or begin a new search, you may enter BEGINn and all previous work will be erased. You will receive time and cost data, and all your set numbering will restart at 1. Another way to change files is by entering .FILEn where n is the number of the file you wish to begin searching in.

If you plan to do several separate searches at one on-line session start new searches with BEGINn as use of .FILEn for several searches may cause you to exceed the 98 set limit which the system has for any one search.
.FILEn is a command best saved for use in related files or for use with the system's SEARCH SAVE feature for executing the same search in several files.
After you begin your search, the system will respond with messages followed by prompts. A "D?" is the search system's prompt, an "I?" is an IIDA prompt. When you receive a prompt, enter the search command which is appropriate or, if you are asked to answer a question or to enter a specific command, respond to the question or directions.
Instructional units are available on the BEGINn command and on file selection.
%00063
Ending the Search
When you have completed your on-line session you will use the signoff command LOGOFF. LOGOFF is the final and irrevocable command which causes disconnection from the information retrieval system. Therefore, you should be certain that you are finished before entering LOGOFF.
The command cannot be abbreviated and may not be stacked with other commands.
The system's response to your LOGOFF command will be the date, time (hours:minutes:second), your user number, and cost and time in the last file used.
The time elapsed (given in decimal hours) is the time for which you are charged for searching since the last timing message. Time messages always accompany the commands BEGINn, .FILEn, and LOGOFF.
$Information is available on the END command which allows you
$to review time and cost information during a search.
$ a (631) END command
$Enter a DIALOG command or "a" if you wish to see the details.
%00071
IIDA Diagnostics
The special diagnostics of IIDA will help you over some rough spots in your search by pointing out syntax errors and suggesting alternative action that will help you make progress toward the search goal. You will get messages automatically when an error is detected on the following:
1. Syntax error - errors in spelling, formatting, and usage of commands.
2. Creation of null sets - you will be reminded when you have created null set
3. Duplicate commands - you will be reminded when you have used essentially the same command to recreate a set
4. Excessive time taken between commands - indicates that you are waiting longer than necessary from receipt of the prompt to enter commands.
5. Unused sets - will remind you of sets created that were not used in combination with other sets to create the final sets
6. Strategy problems - will let you know if you are staying with one approach too long or not long enough to develop the search to its completion.

When these problems are pointed out you will be offered alternatives which you may or may not want to accept. They should serve as suggestions only.
%00111
Full BEGIN
If you enter BEGIN without a file number, the system will respond with requests for information about the searcher, the person for

whom the search is being done and addresses for off-line prints. You will not be using BEGIN in this context at the present time; it an expensive and time-consuming technique used primarily by those who are professional searchers. DO NOT USE THE FULL BEGIN.
%00112

Changing Files

You may wish to change files during the course of your search or to begin a new search during one online session. Begin a new search or enter another file by entering BEGINn or simply Bn where n is the new file you wish to search. Time and cost data will be returned to you and you will be given a prompt to enter the next command of the search. Set numbering will begin at 1. A complete instructional unit is available on FILE SELECTION.
%00121

Expanding an Author's Name

One of the instances in which EXPAND is highly recommended is when you are searching for items by an particular author. Because different works by an author may be indexed by abbreviated first names, initialized first names, or variations of the name, it is important to include all the variations in your search. To EXPAND an author's name, enter EXPAND AU=last name, where "last name" represents the author's last name. If the initials of the first name are known, or if the surname is a common one, you may also enter that information. Note that data bases vary in terms of formatting author's initials and punctuation. (see instructional unit 7 for the particulars.)

*Enter EXPAND AU=LANCASTER
~expand au=lancaster
$Entering EXPAND AU=LANCASTER, for example, yields:

Ref	Index-term	Type	Items	RT
E1	AU=LANAHAN, M. PETER----		1	
E2	AU=LANAHAN, WILLIAM F.--		1	
E3	AU=LANARO, PAMELA-------		1	
E4	AU=LANASA, PHILIP J.----		1	
E5	AU=LANASA, PHILLIP J.---		1	
E6	-AU=LANCASTER----------			
E7	AU=LANCASTER, F. W.-----		6	
E8	AU=LANCASTER, F. WILFRID		1	
E9	AU=LANCASTER, F. WILFRID, ED.----------		4	
E10	AU=LANCASTER, F.W.------		1	
E11	AU=LANCASTER, GEORGE B.-		1	
E12	AU=LANCASTER, H. O.-----		1	
E13	AU=LANCASTER, JOHN------		3	
E14	AU=LANCASTER, JOHN S.---		1	
E15	AU=LANCASTER, JOSEPHAS JACKSON----------		1	
E16	AU=LANCASTER, JOYCE W.--		1	
E17	AU=LANCASTER, LOUISE----		1	
E18	AU=LANCASTER, OTIS E.---		4	

-more-

Note that to find items by F. W. Lancaster you must know that his name is listed under both F. W. and F. Wilfrid in the index. The AU= prefix should be used when searching for works by a particular author, and it is good strategy to first EXPAND the author's name so you may include all variations of the name in your search. All items authored by F.W. Lancaster could now be selected very easily by SELECTing E7-E10.
%00131

SELECTing a Range of Terms

You may save time by entering a range of terms when SELECTing from an

EXPAND table. Instead of entering several SELECT commands, merely specify the range of terms of the EXPAND list by entering SELECT Em-En where m and n represent the first and last E-reference numbers for the list of terms you wish to SELECT.

```
*Enter EXPAND COMPUTER
~expand computer
$Entering EXPAND COMPUTER, for example, yields:
Ref   Index-term                Type Items RT
E1    COMPUTED----------------      575
E2    COMPUTED ASSISTED
        INSTRUCTION-----------        1
E3    COMPUTEER---------------        1
E4    COMPUTEK----------------        1
E5    COMPUTEK MODEL 400-20
        COMPUTER--------------        1
E6   -COMPUTER----------------    11467
E7    COMPUTER ACCESSED
        MICROFICHE LIBRARY----        1
E8    COMPUTER AIDED DESIGN---        3
E9    COMPUTER AIDED DIAGNOSIS        1
E10   COMPUTER AIDED
        INSTRUCTION-----------            1
E11   COMPUTER AIDED
        INSTRUCTION CENTER----        1
E12   COMPUTER AIDED LEARNING-        5
E13   COMPUTER AIDED LEARNING
        PACKAGES--------------        1
                                    -more-

*Enter SELECT E10-E13
~select e10-e13
$Entering SELECT E10-E13, for example, yields:
          1     6  E10-E13
                       E6: COMPUTER
```

You may SELECT more than one range of terms from an EXPAND list by separating the E-reference numbers with commas: for example, you might have selected E4-E6, E8-E9, E10, followed by one carriage return. You may also enter several commands at one time, creating a number of sets at one time. This technique is called stacking.

```
$More detail is available on
$          a (1311) Stacking

$Enter a DIALOG command or a ? if you wish to see this information
?00132
```

SELECTing When Unsure of the Exact Term (Truncation)

Sometimes you may be unsure as to whether to enter a singular or plural form of the term, or of the form a multiple word term takes in the index. Therefore, you may use truncation or the variable-character symbol "?" to select words that begin with the word stem. For example, by entering S LIBRAR? you automatically select terms such as LIBRARY, LIBRARIANS, and LIBRARIES. It is wise to use as long a word stem as possible when using truncation. By cutting off too much of the basic term, you will select many irrelevant terms: in the previous example, selecting LIB? would have also selected such terms as LIBERTY, LIBERTARIAN, and LIBEL.

```
*Enter S COMPUTE?
~s compute?
$Entering S COMPUTE?, for example, yields:
          1  1423 COMPUTE?
```

This response includes items indexed by COMPUTER and COMPUTERS.

The variable character symbol "?" may also be imbedded in terms which can take several forms. For example, you may enter SELECT WOM?N to select WOMEN and WOMAN simultaneously.
%00141
Use of Logical Operators
The logical operators which you will use with the COMBINE command are AND, OR, and NOT. The use of OR will broaden the search by creating a set comprised of items which use either, or both, of the terms of concepts combined. The resultant set may be larger than either of the two sets. Use of AND will narrow the search as its use results in a set which contains items which use both terms or concepts together. The NOT operator narrows the search by excluding concepts or terms from the new set by preventing retrieval of items indexed under the specified NOT condition. NOT used with the COMBINE command means "and not."
*Enter S COMPUTER: S PROGRAMS
^s computer: s programs
^s computer:s programs
^scomputer:sprograms
$Entering S COMPUTER: S PROGRAMS, for example, yields:
6 11467 COMPUTER
7 3867 PROGRAMS
*Enter C 6 OR 7
^ c 6 or 7
^c 6 or 7
^c 6or7
^c6or7
$Entering C 6 OR 7 yields:
8 11856 6 OR 7
*Enter C 6 AND 7
^ c 6 and 7
^c 6 and 7
^c 6and7
^c6and7
$Entering C 6 AND 7 yields:
9 830 6 AND 7
*Enter C 6 NOT 7
^c 6 not 7
^ c 6 not 7
^c 6not7
^c6not7
$Entering C 6 NOT 7 yields:
10 10637 6 NOT 7
Note the differences in set sizes.
%00142
Combined Logical Operations
You may COMBINE logical operations into groups and you may use a range of set numbers to COMBINE several successive sets. When using a combination of operations, parentheses are used to denote which operations are to be performed first. Operations within parentheses are, as in algebra, performed first.
For example, COMBINE(1 OR 2) AND (3 AND 4) NOT 5
will result in the same result as the following 4 commands:
COMBINE 1 OR 2 results in set 6
COMBINE 3 AND 4 results in set 7
COMBINE 6 AND 7 results in set 8
COMBINE 8 NOT 5
Parentheses must be used or logical operations will be performed first on NOT, then on AND and OR logic.
It is possible to COMBINE a range of set numbers with

the same logical operator.

$Detail is available on
$ a (1421) Range feature with COMBINE
$Enter a DIALOG command or a if you wish to see this instructional unit.
?00151
TYPE formats
There are generally eight formats available for viewing the bibliographic information on the citations retrieved. The format in which you will want to see the items will depend on your purpose for seeing the citations.
The formats that yield the full records will give information not only about the document, but also information about the descriptors, identifiers, and other descriptive data which has been assigned to the documents by indexers. This information will be helpful in developing your search strategy; title and abstract formats enable to you review the progress of your search.
$The eight TYPE formats are:
$ a (1511) Format 1 - Accession Number
$ b (1512) Format 2 - Full Record Without Abstract
$ c (1513) Format 3 - Bibliographic Citation with Indexing
$ d (1514) Format 4 - Abstract Only (with accession number)
$ e (1515) Format 5 - Full Record Including Abstract
$ f (1516) Format 6 - Title Only (with accession number)
$ g (1517) Format 7 - Bibliographic Citation Only
$ h (1518) Format 8 - Title and Subject Indexing (Searcher format)
$Enter a DIALOG command or the letter of the format you wish to review.
?00152
Direct TYPE with Accession Number
You may TYPE an item directly if you know its accession number. If, for example, you have browsed through some items in format 6, one item seems particularly relevant, and you wish to TYPE it in a longer format, you may specify the item by its accession number rather than identifying the set number and item number.To direct TYPE, enter TYPE n/f, where n is the accession number and f is the format desired.
You may use this form of TYPE command with any item whose accession number is known to you and you may use any format with the direct TYPE. For example:
*Enter TYPE ED121290/2
~type ED121290/2
$Entering TYPE ED121290/2, for example, yields:
10/2/1
ED121290# IR003267
Impact of On-Line Retrieval Services: A Survey of Users, 1974-75.
Wanger, Judith: And Others
System Development Corp., Santa Monica, Calif.
76 292p.
Sponsoring Agency: National Science Foundation, Washington, D.C.
Available from: System Development Corporation, 2500 Colorado Avenue, Santa Monica, California 90406 (ISBN-8-916368-01-7; $15.00)
Document Not Available from EDRS
Descriptors: Bibliographies/ Computer Programs/ Cost Effectiveness/ Costs/ Data Bases/ Information Centers/ *Information Retrieval/ Information Services / Libraries/ Library Role/ Library Services/ *On Line Systems/ Personnel Needs/ Personnel Selection/ Publicize/ *Use Studies
?00221
Stacking
Instead of entering a long list of SELECTs and COMBINEs, it is

possible to enter several commands in succession, separated by semicolons (;), followed by one carriage return.
*Enter S COMPUTER; S MEMORY; C 1AND2
~s computer; s memory; c1and2
~s computer; s memory; c 1and2
~scomputer;smemory;c1and2
~scomputer; smemory; c1and2
$Entering S COMPUTER; S MEMORY; C 1AND2, for example, yields:
1 54914 COMPUTER
2 6821 MEMORY
3 2234 C1AND2
Notice that the result is the same as if you had made three separate commands, each followed by a carriage return.
Similarly, LIMITing suffixes may be stacked by entering a succession of LIMITing criteria, separated by slash marks (/), followed by one carriage return. Generally you will enter acession number first, followed by major or minor term criteria, and then followed by other restrictions as specified in the individual database files.

The commands which may be stacked are SELECT, COMBINE, and TYPE. BEGIN, END, and LOGOFF may not be stacked.
X00222
Set Range Short Cut
The set range short cut is an abbreviated form for entering several COMBINE commands. To use the set range short cut, enter the COMBINE command in its abbreviated form, the range of set numbers you want to COMBINE, a slash mark (/) and the logical operator by which the sets are to be COMBINEd. NOT may not be used as an operator using this technique.
*Enter s computer; s instruction; s search
~s computer; s instruction; s search
~scomputer; sinstruction; ssearch
~scomputer;sinstruction;ssearch
$Entering S COMPUTER; S INSTRUCTION; S SEARCH, for example, yields:
1 11467 COMPUTER
2 63961 INSTRUCTION
3 3335 SEARCH

Notice that the SELECTs have been stacked.
*Now Enter C1-3/AND
~c1-3/and
$Entering C1-3/OR, for example, yields:
4 59 1-3/AND

The number of sets which use all three terms is 59.
The result is the same as if you had entered each SELECT separately and then had used the longer command:
COMBINE 1 AND 2 AND 3
X00241
Limiting Criteria and Suffixes
The limiting commands require that you set criteria for the limiting condition. The following table summarizes the suffix codes and the criteria by which you may LIMIT sets:

SUFFIX	CRITERIA	FILES
/MAJ	Major descriptor or identifier (The term represents a major topic for the particular citation as indexed and LIMITs the set to those citations in which the search term or concept was used as a major descriptor or	ERIC,NTIS,COMPENDEX

	identifier.)	
/MIN	Minor descriptor or identifier (Used to find citations that would not be found by a manual search of printed indexes - limits the set to those items where the term is not used as a major descriptor or identifier.)	ERIC,NTIS,COMPENDEX
nnnnnn	Accession number By limiting to particular accession numbers you will limit search output to particular years of accession. You will want to review the instructional unit on limiting to particular dates when you need the accession numbers for particular years.	ERIC,NTIS,COMPENDEX
/ED	ED = ERIC DOCUMENT Limits set to educational documents, as apart from journal publications. THIS APPLIES ONLY TO ERIC - File 1.	ERIC
/EJ	EJ = ERIC JOURNAL Limits set to journal articles as apart from other educational documents. THIS APPLIES ONLY TO ERIC - File 1.	ERIC
/AVAIL	Availability Limits set to those publications which are available from EDRS. THIS APPLIES ONLY TO ERIC - File 1.	ERIC
/UNAVAIL	Unavailable Limits set to publications which are unavailable through EDRS. THIS APPLIES ONLY TO ERIC - File 1.	ERIC

There are other means of limiting the size of sets by applying criteria. Use of a SELECT command with suffix codes to limit the search to particular files is discussed in instructional unit 3.

$More detail is available on using the limiting commands with
$ a (2411) Limiting by Accession Number (Date)
$ b (2412) Multiple Limiting
$Enter a DIALOG command or the letter of the entry you wish to see.
X00261
Using Multiple Files
You may wish to repeat a search in more than one file. In order to do this, you must take advantage of the system's search save feature. Otherwise, you would have to repeat all the keying in of commands a second time. The search save feature allows you to save your search formulation for later execution in the same or in another file.
The following steps would be used to execute the same search in multiple files.
1. Begin the search as usual with a begin command.
2. Carry out the search as usual but end it with the command END/SAVE. All your commands will be saved for later execution.
3. Do not, in using search save, select e reference numbers from expand lists, as the e reference numbers will differ in the several files.
4. The system will respond to your end/save command by issuing a unique serial identification number. Save the number - you will need it to execute your search later.
5. Change files by issuing a .FILEn or new BEGINn command.

6. Enter the command .EXECUTEn where n is the serial number of your saved search formulation.
7. The search will be executed in the new file and you may then use the final set in formulating more search strategy. You may also merely type out the results of the saved strategy as used in the second fi
8. After completing the session, enter .RECALL n where n is the serial number the saved search. Then enter .RELEASEn where n is the same serial number. This will erase your saved search from the computer and eliminate storage charges for the search.

YOU MAY NOT USE SEARCH SAVE WHEN USING IIDA. You may wish to use the techniqu later when you use DIALOG without the assistance of IIDA.

%00271

Use of SEARCH SAVE commands

Two new commands are necessary for you to save your search formulation for future use. After beginning your search with the BEGIN command, enter SEARCH SAVE title, where "title" is a title for the search which you assign. This is strictly for your own use. The SEARCH SAVE title will appear as a SELECT command, and you will receive 0 postings. After entering the title for your records, perform your search in the normal way, with two exceptions. First, avoid using time-consuming EXPANDs as they will increase the cost of each subsequent search using the saved search formulation. Secondly, do all SELECTing directly - do not SELECT by using E-reference numbers even if you have had to use EXPAND in the course of the search. E-reference numbers change as new terms are added into the index, and so numbering may change between the orginal search formulation and later execution.

The final command of the search you wish to save should be a COMBINE statement which brings together all the facets you want to include in the search. This is necessary so that when you execute the search later you will retain one set that you can use in further search formulation. Sets other than the final set of the search may not be used in later search formulation. After completing the search, enter the command END/SAVE. This message tells the system to save the search formulation from the END/SAVE command all the way back to the last BEGINn command. The system will respond with a serial identification number. Save the number for use when you are ready to execute the search.

$Details on recalling and executing a saved search are available:
$ a (2711) Recall and Execute Commands
$ b (2712) Releasing the Saved Search
$Enter a DIALOG command or the letter of the entry you want to see.

%00321

Using Prefix Codes

The prefix codes and their use vary accorcing to the file in which you are searching. The prefix code which is used in all the available database files is the author code AU=. To use this prefix, enter the EXPAND or SELECT command, AU=, and the last name of the author. The AU= prefix is used only with personal, not corporate or institutional names.

*Enter EXPAND AU=LANCASTER
~expand au=lancaster
$Entering EXPAND AU=LANCASTER, for example, yields:

Ref	Index-term	Type	Items	RT
E1	AU=LANAHAN, M. PETER----		1	
E2	AU=LANAHAN, WILLIAM F.--		1	
E3	AU=LANARO, PAMELA-------		1	
E4	AU=LANASA, PHILIP J.----		1	
E5	AU=LANASA, PHILLIP J.---		1	
E6	-AU=LANCASTER----------			

```
E7    AU=LANCASTER, F. W.-----      6
E8    AU=LANCASTER, F. WILFRID      1
E9    AU=LANCASTER, F.
        WILFRID, ED.----------      4
E10   AU=LANCASTER, F.W.------      1
E11   AU=LANCASTER, GEORGE B.-      1
E12   AU=LANCASTER, H. O.-----      1
E13   AU=LANCASTER, JOHN------      3
E14   AU=LANCASTER, JOHN S.---      1
E15   AU=LANCASTER, JOSEPHAS
        JACKSON---------------      1
E16   AU=LANCASTER, JOYCE W.--      1
E17   AU=LANCASTER, LOUISE----      1
E18   AU=LANCASTER, OTIS E.---      4
                                 -more-

You may SELECT a range of terms to pick up variations of the author's name.
$Details are available on the prefix codes applicable in three databases:
$          a (3211) ERIC
$          b (3212) NTIS
$          c (3213) COMPENDEX
$Enter a DIALOG command or the letter of the entry which you wish to see.
?00322
Use of Suffixes
The subfiles of the basic file which can be searched by specifying a
suffix code include the following:
Field                    Suffix Code          Files
-----                    -----------          -----
Title                    /TI                  ERIC,NTIS,COMPENDEX
Abstract                 /AB                  ERIC
Descriptor               /DE                  ERIC,NTIS,COMPENDEX
Full Descriptor          /DF                  ERIC,NTIS,COMPENDEX
Primary Descriptor       /DE*,/DF*            ERIC,NTIS,COMPENDEX
Identifier               /ID                  ERIC,NTIS,COMPENDEX
Full Identifier          /IF                  ERIC,NTIS
Primary Identifier       /ID*,/IF*            ERIC,NTIS
Descriptive Note         /NT                  ERIC
Corporate Source         /CS                  ERIC,NTIS
Author Affiliation       /CS                  COMPENDEX

To search in the title field, for example, enter the SELECT command
with the term of interest, and add the suffix to designate the
title field.
*Enter S COMPUTER/TI
?s computer/ti
$Entering S COMPUTER/TI, for example, yields:
          1  2543 COMPUTER/TI
$Definitions of the suffix-coded subfiles are available:
$          a (3221) Title /TI
$          b (3222) Abstract /AB
$          c (3223) Descriptor /DE,/DF,/DE*,/DF*
$          d (3224) Identifiers /ID,/IF,/ID*,/IF*
$          e (3225) Corporate Source & Author Affiliation /CS
$          f (3226) Descriptive Note /NT
$Enter a DIALOG command or the letter of the entry you wish to see.
?00323
Use of Multiple Suffixes
More than one searchable field may be specified by the use of suffixes,
and the list of fields may be specified in a single command.  For example,
the command SELECT COMPUTERS/TI will retrieve items with 'computers'
```

in the title, but if you want to retrieve items with the term used in the abstract as well you may enter SELECT COMPUTERS/TI,AB and follow the command with one carriage return. Each successive suffix you add will broaden the search results: if you were to specify all fields with suffixes the result would be the same as if you had specified no suffixes - the searching will be done in the basic index (the merged file which includes all suffix-coded subfiles).

```
*Enter SELECT COMPUTERS/TI
~select computers/ti
$Entering SELECT COMPUTERS/TI, for example, yields:
          1   412 COMPUTERS/TI
*Enter SELECT COMPUTERS/TI,AB
~select computers/ti,ab
$Entering SELECT COMPUTERS/TI,AB yields:
          2  2054 COMPUTERS/TI,AB
```

Notice that the suffixes are separated by a comma when using multiple suffixes.

Remember that searchable fields are specified within each database file, so do not attempt to search in an unsearchable field. The details of database-dependent suffixes are available in the instructional unit on "the use of suffixes" and "database descriptions."

%00331

Infixes and Their Use

The following table lists the infixes and the definition of the use of each. An example of a select command using the infixes is included for each.

Infix symbol	When used, requires:	Example
(C)	both words to appear anywhere in the citation: using (C) is equal to combining the two terms with "and" logical operator and COMBINE command	s solar(c)energy
(F)	terms to be in the same field: any suffix-coded field: terms may appear in any order	s solar(f)energy
(W)	terms to be directly adjacent: in any suffix coded field:must be in the order specified in the SELECT command	s solar(w)energy
(nW)	terms to be in specified order with n or fewer terms between: any suffix-coded field (n=number of intervening terms which may be allowed)	s solar(4w)energy
(L)	heirarchical relationship of the terms within the descriptor field: database-dependent notation (only COMPENDEX, of the IIDA available databases, uses the (L) infix).	

Truncation may not be used with statements restricted by full-text operators. There are 12 words, which are called "stop words" which are not included in the indexes and which should not be used as search terms. These are : a, an, and, by, for, from, in, of, on, the, to, and with. Where these words occur between two terms, as part of a

multiple-word term you wish to search on, denote the stopword by using the infix (nW) where n is 1 if there is one stop word between the two terms. For example, if a multiple word term "storage and retrieval" were of interest, you would enter it as STORAGE(1W)RETRIEVAL, not as STORAGE AND RETRIEVAL.
Full text operators are used with the SELECT command. You may use suffixes to denote the field in which the multiple word term is to be searched. For example, you would enter SELECT ACQUISITION(F)SOFTWARE/TI to find items with the two terms in any order in the title field.
%00511
ERIC Search Methods and Formats
Subject or text searching in File 1 may be accomplished by searching the following fields which are included in the basic index:
abstract, /ab
descriptor, /de
corporate source, /cs
identifier, /id
descriptive note, /nt
sponsoring agency, /sa
title, /ti.

You may also search primary terms (denoted with an asterisk) as in /df*, /if*, /id* and /de*. Single word descriptors, used by themselves rather than merely in the descriptor field, are denoted by the suffix /df; these are full descriptors. The /if suffix denotes full identifiers.
Prefix codes fields in ERIC include:
Area code (legislative), ac=
personal author, au=
clearinghouse code, ch=
contract/grant number, cn=
document type, dt=
group code, gc=
issue, is=
journal name, jo=
report number, rn=
corporate source or sponsoring agency, sc=
update, ud=
year, yr=.

Limiting is used in file 1 ERIC with the following suffix codes:
/ed, limits to particular accession numbers and/or ED (nonjournal) subfile
/ej, limits to particular accession numbers and/or EJ (journal) subfile
/avail, limits items retrieved to items available from ERIC EDRS
/unavail, limits to documents not available from ERIC,EDRS
/maj, limits to major descriptors or identifiers
/min, limits to minor descriptor or identifier field.

TYPE formats available in ERIC are formats 1 through 8.
%00512
Expanding Related Terms
You may EXPAND the terms from a previous EXPAND-list if the terms have at least one related term (RT). The system will respond with subject-related terms. To expand from the previous EXPAND-list, enter EXPAND En or EEn where n represents the E-reference number of the term you have chosen.
*Enter EXPAND E6
~expand e6
$Entering EXPAND E6, for example, yields:

Ref	Index-term	Type	Items	RT
R1	INFORMATION RETRIEVAL---		2191	22
R2	SEARCH STRATEGIES-------N		424	10
R3	INFORMATION UTILIZATION-B		1194	14
R4	BIBLIOGRAPHIC COUPLING--R		32	11
R5	CITATION INDEXES--------R		102	6
R6	CODIFICATION------------R		293	6
R7	COMPUTATIONAL LINGUISTICS-----------R		373	12
R8	COORDINATE INDEXES------R		52	10
R9	DATA PROCESSING---------R		1128	22
R10	DIAL ACCESS INFORMATION SYSTEMS---------------R		133	17
R11	DOCUMENTATION-----------R		1898	24
R12	INDEXES (LOCATERS)------R		1176	29
R13	INDEXING----------------R		1149	17
R14	INFORMATION NEEDS-------R		1344	8
R15	INFORMATION SEEKING-----R		488	12
R16	INFORMATION SERVICES----R		1491	15
R17	INFORMATION SYSTEMS-----R		2689	21
R18	INFORMATION THEORY------R		729	19

-more-

X00513
Expanding Terms Directly to Subject-Related Term
It is possible to EXPAND a term directly to subject-related lists rather than examining the alphabetically-related terms and then EXPANDing from that list. To EXPAND directly, enter EXPAND (term) or E (term) where "term" is the term of interest and is enclosed in parentheses. The response will be a subject-related list.
*Enter EXPAND (INFORMATION RETRIEVAL)
~expand (information retrieval)
$Entering EXPAND (INFORMATION RETRIEVAL), for example, yields:

Ref	Index-term	Type	Items	RT
R1	INFORMATION RETRIEVAL---		2191	22
R2	SEARCH STRATEGIES-------N		424	10
R3	INFORMATION UTILIZATION-B		1194	14
R4	BIBLIOGRAPHIC COUPLING--R		32	11
R5	CITATION INDEXES--------R		102	6
R6	CODIFICATION------------R		293	6
R7	COMPUTATIONAL LINGUISTICS-----------R		373	12
R8	COORDINATE INDEXES------R		52	10
R9	DATA PROCESSING---------R		1128	22
R10	DIAL ACCESS INFORMATION SYSTEMS---------------R		133	17
R11	DOCUMENTATION-----------R		1898	24
R12	INDEXES (LOCATERS)------R		1176	29
R13	INDEXING----------------R		1149	17
R14	INFORMATION NEEDS-------R		1344	8
R15	INFORMATION SEEKING-----R		488	12
R16	INFORMATION SERVICES----R		1491	15
R17	INFORMATION SYSTEMS-----R		2689	21
R18	INFORMATION THEORY------R		729	19

-more-

X00521
Search methods and formats in NTIS
Searching may be done in the basic index which includes corporate source (/cs), descriptor (/de), identifier (/id), and title fields (/ti). Subfields which are suffix - coded are searchable as are major

descriptors (/de*), full (single word) descriptors (/df*), major identifiers (/id*), full identifiers (/if), major identifiers (/id*), and major full identifiers (/if*).
Prefix-coded fields include: personal author (au=), corporate source code (cc=), subject category field (cf=), contract number (cn=), contract number prefix (cp=), journal announcement (ja=), report number (rn=) and update (ud=).
Limiting (LIMIT and LIMITALL) is restricted to accession number, major descriptors, major identifiers (/MAJ) or all occurances except major identifiers and descriptors (/MIN).
Direct access to a particular item is available by using simply the accession number in conjunction with the TYPE command and the desired format.
Formats available in NTIS are 1 (DIALOG accession number), 2 (full record without abstract), 3 (bibliographic citation), 4 (NTIS report or accession or order number), 5 (full record) 6 (short citation).

It should be noted that abstracts are available in NTIS items, but the abstract field is not a searchable field.
$More detail is available on code searching in NTIS:
$ a (5211) Code searching in NTIS
$Enter a DIALOG command or "a" to see the additional details.
%00531
Search methods and formats in COMPENDEX

Searchable fields in COMPENDEX include the basic index, comprised of abstract, corporate source, descriptor, identifier, and title fields. Major descriptor, full descriptor, and major full descriptor fields (/df, //de*, and /df*) are also searchable in COMPENDEX.
Code searching in the following fields may be done in COMPENDEX: author (au=), card-alert codes (ca=), CODEN (co=), journal announcement (ja=), and update (ud=).

Limiting in COMPENDEX is confined to accession number, major heading or subheading (/ma]), or all occurances except major heading or subheading (/min

Formats available in COMPENDEX include format 1 (DIALOG accession number), format 2 (full record except abstract), format 3 (Bibliographic citation), format 5 (full record), and format 6 (title, source, and EI abstract number).
$More detail is available on :
$ a (5311) Code searching in COMPENDEX
$Enter a DIALOG command or "a" if you wish to see the detailed information.
%00631
END command
The END command may be used during a search if you wish to see time and cost data for your time in that file in which you are working. Set numbering will not be restarted at 1, but time, cost, and file number will be returned. You may return to the search following an END command.
%01311
Stacking SELECTs
To save time you may SELECT directly more than one term at a time. This is accomplished by entering the terms with the SELECT commands on one line, separating them by semicolons (;). You may SELECT any number of terms as long as they fit on one line.
*Enter SELECT INFORMATION RETRIEVAL; SELECT INTERMEDIARIES
*select information retrieval; select intermediaries

```
          2  2191 INFORMATION RETRIEVAL
          3  498 INTERMEDIARIES
Both sets can now be used in future set-creating steps of the search.
%01421
Range feature with COMBINE
When you want to COMBINE several successive set numbers, such as
combining sets 1,2,3,and 4 with the OR logical operator, you can
use a short-cut to enter the commands.  Instead of using the
longer command(COMBINE 1 OR 2 OR 3 OR 4) you may enter C1-4/OR
where 1 to 4 is the range of sets you want to COMBINE and OR
is the appropriate logical operator.
%01511
Format 1
Format 1 gives the accession number of the document retrieved.
For example:
*Enter TYPE 10/1/1
~type 10/1/1
$Entering TYPE 10/1/1, for example, yields:
10/1/1
ED121290
%01512
Format 2
Format 2 is the format for the full record, with the exception of
the abstract of the document. Specify Format 2 when you wish to
see not only the bibliographic information, but also the list of
descriptors and identifiers for use in future search strategy
formulation.  For example:
*Enter TYPE 10/2/1
~type 10/2/1
$Entering TYPE 10/2/1 yields:
10/2/1
ED121290#  IR003267
  Impact of On-Line Retrieval Services: A Survey of Users, 1974-75.
  Wanger, Judith: And Others
  System Development Corp., Santa Monica, Calif.
  76   292p.
  Sponsoring Agency: National Science Foundation, Washington, D.C.
  Available from:  System Development Corporation,  2500 Colorado
Avenue, Santa Monica, California 90406 (ISBN-8-916368-01-7: $15.00)
  Document Not Available from EDRS
  Descriptors: Bibliographies/ Computer Programs/  Cost Effectiveness/
Costs/  Data Bases/  Information Centers/ *Information Retrieval/
Information Services / Libraries/ Library Role/ Library Services/  *On
Line Systems/ Personnel Needs/  Personnel Selection/  Publicize/  *Use
Studies
%01513
Format 3
Format 3 is the format to specify when you want to see
the bibliographic citation for the item.  This will give you
title, author, date of publication, journal or document title,
and other information you will need to locate the item after you
have completed your search.  For example:
*Enter TYPE 10/3/1
~type 10/3/1
$Entering TYPE 10/3/1 yields:
10/3/1
ED121290#
  Impact of On-Line Retrieval Services: A Survey of Users, 1974-75.
  Wanger, Judith: And Others
  System Development Corp., Santa Monica, Calif.
```

292p. 76
System Development Corporation, 2500 Colorado Avenue, Santa Monica, California 90406 (ISBN-8-916368-01-7: $15.00)
Document Not Available from EDRS
X01514
Format 4
Format 4 is the format which gives the accession number and the document's title and abstract. The abstract is the description of the document, designed to let you know what kinds of information you can expect to find in the document. The abstract may vary from a very few words to a very few long and thorough description. Format 4 should be specified when you want to see if the items retrieved meet your search needs. For example:
*Enter TYPE 10/4/1
~type 10/4/1
$Entering TYPE 10/4/1 yields:
10/4/1
ED121290#
Impact of On-Line Retrieval Services: A Survey of Users, 1974-75.
Over 1,250 users of 10 major on-line bibliographic information retrieval services were surveyed to assess the impact of such services on the library and information science community and its user community. The major topics considered in analyzing responses were: (1) respondent backgrounds; (2) methods of acquiring and promoting on-line services; (3) selection and training of staffs; (4) levels of use; (5) selection, access, and use of on-line systems; (6) selection and use of data bases; (7) costs; (8) major problems; and (9) major areas of impact. It was found that the services are being used primarily by libraries and information service centers in commercial organizations, universities, and federal government agencies. Searchers are primarily information intermediaries--librarians and information specialists. The use of search services is generally considered cost-effective. The steady growth of these services will mean increased pressure for the development of the following services: (1) accommodation for a wide range of user expertise within the search systems; (2) new methods and wider availability of user training; (3) methods for recovering the costs of retrieval service withing traditionally no-fee institutions; and (4) faster methods of acquiring full-text copies of materials indexed. (Author/PF)
X01515
Format 5
Format 5 gives the full record of bibliographic information, full indexing information, and the document's abstract as well. Format 5 is the most complete format you can use with the TYPE command. For example:
*Enter TYPE 10/5/1
~type 10/5/1
$Entering TYPE 10/5/1 yields:
10/5/1
ED121290# IR003267
Impact of On-Line Retrieval Services: A Survey of Users, 1974-75.
Wanger, Judith: And Others
System Development Corp., Santa Monica, Calif.
76 292p.
Sponsoring Agency: National Science Foundation, Washington, D.C.
Available from: System Development Corporation, 2500 Colorado Avenue, Santa Monica, California 90406 (ISBN-8-916368-01-7: $15.00)
Document Not Available from EDRS
Over 1,250 users of 10 major on-line bibliographic information retrieval services were surveyed to assess the impact of such services

on the library and information science community and its user community. The major topics considered in analyzing responses were: (1) respondent backgrounds; (2) methods of acquiring and promoting on-line services; (3) selection and training of staffs; (4) levels of use; (5) selection, access, and use of on-line systems; (6) selection and use of data bases; (7) costs; (8) major problems; and (9) major areas of impact. It was found that the services are being used primarily by libraries and information service centers in commercial organizations, universities, and federal government agencies. Searchers are primarily information intermediaries--librarians and information specialists. The use of search services is generally considered cost-effective. The steady growth of these services will mean increased pressure for the development of the following services: (1) accommodation for a wide range of user expertise within the search systems; (2) new methods and wider availability of user training; (3) methods for recovering the costs of retrieval service within traditionally no-fee institutions; and (4) faster methods of acquiring full-text copies of materials indexed. (Author/PF)

Descriptors: Bibliographies/ Computer Programs/ Cost Effectiveness/ Costs/ Data Bases/ Information Centers/ *Information Retrieval/ Information Services / Libraries/ Library Role/ Library Services/ *On Line Systems/ Personnel Needs/ Personnel Selection/ Publicize/ *Use Studies

%01516

Format 6

Format 6 is the Title format, and the response to a command to TYPE in format 6 consists of the accession number and the document's title. Sometimes you may want to use this format to determine if titles appear to meet your search specifications, and also to make a less time-consuming judgment of your search performance before deciding to TYPE out the longer records. For example:

*Enter TYPE 10/6/1
~type 10/6/1
$Entering TYPE 10/6/1 yields:
10/6/1
ED121290#
Impact of On-Line Retrieval Services: A Survey of Users, 1974-75.

%01517

Format 7

Format 7 is called the "end user" format in that it gives all bibliographic information and the abstract, but it excludes subject indexing information. The list of descriptors and identifiers, that are assigned the document by indexers, are excluded in Format 7. For example:

*Enter TYPE 10/7/1
~type 10/7/1
$Entering TYPE 10/7/1 yields:
10/7/1
ED121290# IR003267
Impact of On-Line Retrieval Services: A Survey of Users, 1974-75.
Wanger, Judith; And Others
System Development Corp., Santa Monica, Calif.
76 292p.
Sponsoring Agency: National Science Foundation, Washington, D.C.
Available from: System Development Corporation, 2500 Colorado Avenue, Santa Monica, California 90406 (ISBN-0-916368-01-7: $15.00)
Document Not Available from EDRS

Over 1,250 users of 10 major on-line bibliographic information retrieval services were surveyed to assess the impact of such services on the library and information science community and its user

community. The major topics considered in analyzing responses were: (1) respondent backgrounds; (2) methods of acquiring and promoting on-line services; (3) selection and training of staffs; (4) levels of use; (5) selection, access, and use of on-line systems; (6) selection and use of data bases; (7) costs; (8) major problems; and (9) major areas of impact. It was found that the services are being used primarily by libraries and information service centers in commercial organizations, universities, and federal government agencies. Searchers are primarily information intermediaries--librarians and information specialists. The use of search services is generally considered cost-effective. The steady growth of these services will mean increased pressure for the development of the following services: (1) accommodation for a wide range of user expertise within the search systems; (2) new methods and wider availability of user training; (3) methods for recovering the costs of retrieval service within traditionally no-fee institutions; and (4) faster methods of acquiring full-text copies of materials indexed. (Author/PF)

X01518

Format 8

Format 8 is called the "searcher" format in that it gives only the title and subject indexing for the retrieved documents. The list of descriptors and identifiers can be helpful to you when you know of a title and you are searching for documents "like" the one you are familiar with. By printing out the descriptors and identifiers assigned to the known document you can borrow the indexing terms to formulate your strategy to find similar items in the file.

For example:

*Enter TYPE 10/8/1

~type 10/8/1

$Entering TYPE 10/8/1 yields:

10/8/1

ED121290#

Impact of On-Line Retrieval Services: A Survey of Users, 1974-75.

Descriptors: Bibliographies/ Computer Programs/ Cost Effectiveness/ Costs/ Data Bases/ Information Centers/ *Information Retrieval/ Information Services / Libraries/ Library Role/ Library Services/ *On Line Systems/ Personnel Needs/ Personnel Selection/ Publicize/ *Use Studies

X02411

Limiting by Accession Number (Date)

When using the DIALOG system you may LIMIT the retrieved citations by the date the items were accessed into the file. However, to do this the limiting criteria is accession number, not date of publication or accession. Each database differs in terms of which accession numbers correspond to particular years. The following table will give you the first accession number for each year since 1970. When you wish to limit your search to publications added to the file since any of the years since 1970, simply enter LIMITn/and the range of accession numbers from the table. (n is the set number you wish to restrict.) The range is specified as, for example, <<100001-200001>>, with the numbers separated by a hyphen.

Date	ERIC-ED	ERIC-EJ	COMPENDEX	NTIS
1970	031605	011708	000001	6875L2
1971	042061	027600	100001	A1193H4
1972	054391	045272	200001	A322414
1973	066621	062753	300001	A5551E1 (NON-AEC)-C0011A1
1974	080788	082165	400001	A6171H1-C1941I3
1975	095254	101873	400001	A6715I1 - C3824I4

1976 110595 121927 500001 A7185H2 -C5571C1
To update from a particular year to the present time, enter the accession number given in the table for the first year and then make the upper limit of the range a "nine-fill" figure according to the following table:

DATABASE	Nine-fill format
ERIC	use 999999/ED or 999999/EJ as upper limit
NTIS	use A9991A1 for ERDA reports without abstracts; use D9991A1 for all others.
COMPENDEX	use 999999 as upper limit

%02412
Multiple Limiting
It is possible to LIMIT one set with several LIMITing criteria. You may accomplish this most simply by applying each successive limit to the result of the previous criteria. This technique will reduce the size of the set considerably.
For example, you may LIMIT a set to major terms by entering
LIMIT 10/MAJ
which will result in a new set (no.11). Then, if the set is still too large, you might limit it to items accessed by the database in the most recent three years by entering
LIMIT 11/110595-999999
This technique of adding criteria for limiting will reduce set size rapidly.
%02711
Recall and Execute Commands
The .RECALL command is used to recall, but not to execute, the saved search formulation. To use the command, enter .RECALLnnn where "nnn" represents the three-character serial identification assigned to your saved search formulation.
After you use the .RECALL (notice the period (.) preceeding the word "RECALL") command, you may execute or release the saved search.
The .EXECUTE command is used to carry out the saved search in the file of your choice. To use this command, you must first enter a period (.) and then the word EXECUTE and the three character serial identification.
Use of the .EXECUTE command causes all steps in the search to be carried out, but the only set which will be returned is the final set, the one created by the final COMBINE command of the search.
The system response to the .EXECUTE command is a set number, the number of items in that set, and the serial number of the saved search. The set number of the final set may then be used further in developing the search in more detail or breadth.
$More information is available on executing the search in another file.
$ a (27111) Executing the Saved Search in Another File
$Enter a DIALOG command or "a" if you wish to see the detailed information.
%02712
Releasing the Saved Search
The .RELEASE command is used to erase the saved search when it is no longer worthwhile to save it in the system. The saved search must be recalled, and then released. The sequence of commands used to release a saved search from storage is:
1. Enter .RECALLnnn where nnn is the serial identification of the saved search.
2. When the saved search formulation has been printed out, enter .RELEASEnnn where nnn is the serial identification number of the search you are releasing from storage.

The system will respond with a repeat of your .RELEASE

command and the serial identification.

X03211

ERIC - File 1

The following table lists the data field, prefix codes, and examples of use for the ERIC file.

Data Field	Prefix Code	Example
Area Code (Legislative)	AC=	AC=32
Personal Author	AU=	AU=Smith, J
Contract or Grant Number	CN=	CN=OE-2431
Clearinghouse Code	CH=	CH=EC
Document Type	DT=	DT=8
Sponsoring Agency	SC=	SC=AHP008956
Journal Name	JO=	JO=AAUP BULLETIN
Project Number	PN=	PN=H-356897
Report Number	RN=	RN=TM/3098
Update	UD=	UD=7709
Year of Publication	YR=	YR=78

The abbreviations, project and code numbers, and agency data will be known to you if you are searching for particular reports or journal articles, etc.

X03212

NTIS - File 6

The following table lists the data field, prefix codes, and examples for use for the NTIS file.

Data Field	Prefix Code	Example
Personal Author	AU=	AU=Jones, J.F.
Corporate Source Code	CC=	CC=509008
Subject Category Field	CF=	CF=9
Contract Number	CN=	CN=AF-2390
Contract Number Prefix	CP=	CP=AF
Journal Announcement	JA=	JA=D78
Report Number	RN=	RN=MTR-396
Report Number Prefix	RP=	RP=MT
Update	UD=	UD=78

The abbreviations, code numbers, dates, and subject categories will be known to you when you begin your search, or you may see the instructional unit on individual database files for more information.

X03213

COMPENDEX - File 8

The following table lists the prefix codes and the corresponding fields for which they stand, along with an example of how the codes are used.

Field Name	Prefix Code	Example
Personal Author	AU=	AU=Brown, John M.
Journal Code	CO=	CO=(6 character CODEN)
Card Alert Code	CA=	CA=43
Update	UD=	UD=77
Journal Announcement	JA=	JA=D78

Abbreviations, code numbers, and dates for prefix use should be known to you when you plan your search; for more particulars, you may review the instructional unit on the individual database file or refer to the printed guide.

X03221

Title /TI

Specifying the title field means that citations retrieved will have the term of interest in the title of the document.

The /TI suffix can be used in ERIC,NTIS, and COMPENDEX.
%03222
Abstract /AB
The abstract field may be searched in the ERIC file. When you use the /AB suffix, the search is conducted in the abstract field and the term specified will appear in the descriptive summary of the document.
%03223
Descriptors
Descriptors are words or phrases which describe the subject of the document. These terms are assigned to the document by indexers and are terms which describe the subject matter of the document in specific terms. Each document in the file is assigned several descriptors - the number depends on the document, the file, and the extent to which the document is indexed.
The /DE suffix denotes that the term is to be found in the descriptor field - it may be a single word descriptor or part of a multiple word term. /DF stands for FULL DESCRIPTOR and means that the word or term is not part of a multi-word descriptor, but stands alone.
The /DE and /DF suffixes may be used in all the files which may be searched through IIDA.
If the /DE or /DF suffix is followed by an asterisk (*), the descriptors searched will be those considered by the indexers to be the primary terms which describe the item. Often the (*) terms correspond to the index terms used as subject headings in printed indexes to the literature. Using the /DE* or /DF* suffix further limits the search output and may be used in all IIDA database files.

The descriptor field is the field which is searched when you use a multiword thesaurus term : this is automatic, and therefore makes it unnecessary to specify a suffix unless the several words are linked by infixes.
%03224
Identifiers
Identifiers are terms which describe the contents of the documents. They are not terms which describe the subject matter of the documents. They may be specified as words or phrases used alone (/IF) or a part of a larger multi-word term (/ID).
An asterisk (*) following the /ID or /IF suffix dentotes that the indexer of the document felt that the term was the primary indicator of a description of the document apart from its subject matter.
The /ID suffix may be used in all IIDA database files: /IF, /ID*, and /IF* are searchable only in ERIC and NTIS.
%03225
Corporate Source & Author Affiliation
The /CS suffix denotes that the term you are searching on is to appear in the corporate source or author affiliation field. Names of schools, institutions, corporations, and others who sponsor research may be included. You may, for example, search on 'Hopkins' followed by the /CS suffix when you wish to find research which originated at Johns Hopkins University.
The /CS suffix denotes corporate source in ERIC and NTIS, and author affiliation in COMPENDEX.
%03226
Descriptive Note /NT
A descriptive note field, designated by the suffix /NT,is searchable in the ERIC file only. The descriptive note includes information about the circumstances of publication, language of document, pagination, and other distinguishing information which sets one document apart from another.
This is not to be confused with the /AB field: /NT is a suffix

used only in ERIC and applies there only to non-journal items.
?05211
Code Searching in NTIS
The prefix codes in NTIS are listed below with procedure for searching:
AUTHOR (AU=)
To search on an author's name, enter the command, the au= prefix, and the author's surname, followed by a comma, followed by a space, followed by the first name or first initial.
Expansion of the author's surname is recommended.
SUBJECT CATEGORY FIELD (CF=)
Two types of subject category fields appear here: the COSATI code and the NTIS code. These codes are used in conjunction with arranging and classification of reports. The code numbers consist of two numerics and one alpha character. The printed Guide to DIALOG Databases lists the COSATI code numbers that are appropriate for searching in various subject areas.
COPRORATE SOURCE CODE (CC=)
This field consists of the six-digit Defense Documentation Center (DDC) source codes, or the seven-digit ERDA corporate source code numbers. NTIS uses all DDC codes in indexing documents.

CONTRACT NUMBER (CN=)
The contract number prefix may be used for searching contract, project, grant, and monitor numbers. These numbers may contain numbers, letters, hyphens, or slash marks. It is suggested that the numbers be expanded to find the proper format.
CONTRACT NUMBER PREFIX (CP=)
The contract number prefix code is used to collect all grant, contract, task, or monitor numbers by the characgter string which appears before the first hyphen or slash in the contract number.
For example, SELECT CP=NSF would retrieve all items of work sponsored by the National Science Foundation (prefix NSF).
REPORT NUMBER (RN=)
The report number is used to collect reports into various groupings. The report number prefix codes (RP=) as well as the RN= codes are explained in the printed Guide to DIALOG databases.
JOURNAL ANNOUNCEMENT (JA=)
This field is used to indicate volume and issue of the journal in which the report was first announced. Prefix letters that are used for each journal are listed in the printed GUide to DIALOG Databases.
?05311
Code Searching in COMPENDEX
The following are the prefixes which you can use to search fields apart from the basic index.
AUTHOR (AU=)
Up to 16 personal authors may be listed per document. The au= prefix should be used to search for work by a particular author. First enter the EXPAND or SELECT command, then the prefix au=, then the author's surname, a comma, and finally the first name, names, or initial. In COMPENDEX, an "e" is added for the umlaut in German names.
It is reccmmended that personal author names be EXPANDed prior to SELECTing, as there may be several variations of the name in the file.

CODEN (CO=)
The CODEN is the standard five-digit code which provides a unique identification of a periodical title. Prior to 1972, the five digit code was used, but a sixth check character has now been added. Therefore, it is recommended that truncation be used with the CODEN, as SELECT CO=JACTA?
CODENs may be found in the printed Publications Indexed for Engineering.
The CODEN may be used in the search to include or exclude specific periodicals

as sources of documents in the search.

JOURNAL ANNOUNCEMENT (JA=)
The information in this field is the year and month the document appeared in the printed EI Monthly. For example SELECT JA=7711 for all documents which appeared in the November 1977 issue of EI Monthly.

UPDATE (UD=)
Search results can be restricted to specific updates of the database. For example, SELECT UD=7702 restricts results to the particular update from 1977. In addition, SELECT UD=9999 will retrieve the latest update.

CARD-A-LERT CODES (CA=)
Card-A-Lert Codes, since they represent a broad classification scheme, are useful in searching the COMPENDEX file. Although the Card-A-Lert current awareness service was discontinued, the three-digit codes as defined in the EI users manual may be used for searching.
Valid Card-A-Lert codes are three digit numbers above 400 which end in a non-zero digit.
You may search the first two digit of the Card-A-Lert code - this will retrieve all related subject areas. For example, SELECT ACCELERATION(C)CA=43 will restrict the occurance of "acceleration" to the concept of "transportation" or "trarsportation vehicles" as defined by Card-A-Lert code 431-434.

$More information is available on
$ a (53111) Special characters in COMPENDEX
$Enter a DIALOG command or "a" if you wish to see the detailed information.
%27111
Executing the Saved Search in Another File
At times you will want to execute one search in several files to be certain you have collected a comprehensive set of citations to meet your needs. SEARCH SAVE allows you to do this without having to reenter all the commands or reformulate your search strategy. When you have used the END/SAVE command to save a search you may then change files by entering .FILEn where n is the number of the file to which you are changing. Then follow the procedure for recalling and/or executing the search. If you want to see the formulation, to be certain your formulation is the one for your present need, use the .RECALL command before executing the search. You may also use .EXECUTE without having recalled the search first.
The result of execution of the search formulation in a new file will be a final set number and the number of postings. You may TYPE out a few of the items to be certain the items are of interest. You may continue to refine the search or browse through other citations to find relevant items.
It is wise to beware when changing files for execution of a search: files differ in terms used in the index and with regard to the documents included in the database. You should also note that certain citations may be retrieved in both files as there is a certain amount of overlap between some of the files.
%53111
Special characters in COMPENDEX
When searching in COMPENDEX it is necessary to be familiar with some special characters. For example, formulas which appear in titles and abstracts are set off in the basic index by special markings. Each superscript character is preceeded by two asterisks (**), as 10**1**8 and is retrieved by SELECT 10 (W)1(W)8
Similarly, each subscript character is preceded by two slashes, e.g.

CO//2 which is retrieved by SELECT CO(W)2

Additionally, the appearance of the left bracket phrase generally indicates the presence of an English translation of a foreign title. To separate English from non-English titles, therefore, "left braket" should be selected from the title field using full-text methods, as SELECT left(w)bracket/ti. If you are searching strictly for English language documents (not in translation) the "left bracket" items can be dropped by using the NOT operator with the COMBINE command. You may want to refer to the printed guide to DIALOG databases for more information.

INDIVIDUALIZED INSTRUCTION FOR DATA ACCESS (IIDA)

Quarterly Report No. 6
September, 1979

Drexel University, School of Library
and Information Science
and the Franklin Research Center

NSF Grant No. DSI-77-26524

IR008572

TABLE OF CONTENTS

	Page
1. Introduction	3
2. Conit and Iida	4
3. Control Programs	5
3.1. Executable Command Segment	5
3.2. Pre-search Initialization	7
3.2.1. get_user_id.pll	7
3.2.2. db_init.pll	8
3.2.3. student.incl.pll	8
3.3. Main procedure	10
3.3.1. iida.pll	10
3.3.2. ttstart	13
3.4. Post-search reporting	16
3.4.1. proc_rep.pll	16
3.4.2. reports.pll	18
3.4.3. adapt_rep.pll	19
4. Search-Oriented Programs	20
4.1. Network and DIALOG communication	21
4.1.1. netcon.pll	21
4.1.2. callout.pll	22
4.1.3. sendline.pll	22
4.1.4. getline.pll	22

	Page
4.2. Command and Response Parsing	23
4.2.1. parse.pl1	23
4.2.2. rsvparse.pl1	26
4.2.3. idverb.pl1	27
4.3. Command and Response Analyzing	28
4.3.1. canalysis.pl1	28
4.3.2. normalizer.pl1	29
4.3.3. ranalysis.pl1	30
5. Other Support Programs	32
5.1. Exercises local to MULTICS	32
5.1.1. ex1.pl1	33
5.1.2. ex3.pl1	33
5.2. User-Initiated Assistance	34
5.2.1. help_iida.pl1	34
5.2.2. ex3.pl1 (as help)	35
5.2.3. da.pl1	36
5.2.4. rec_assist.pl1	37
5.3. Message Handlers	40
5.3.1. messen.pl1	40
5.3.2. mesyner.pl1	40
5.3.3. mess_out.pl1	41
Appendix 1: Sample Proctor Report	1-1
Appendix 2: Sample Adaptive Variable Report	2-1
Appendix 3: Base Messages for Syntactic Errors	3-1
Appendix 4: List of Strategic Conditions by Code	4-1

1. INTRODUCTION

The present report intends to describe the current state of all computer programs that make up the Individualized Instruction for Data Access (IIDA) system. This report will be technical and program-oriented. The function and operation of each significant program in the system will be discussed in turn.

Certain assumptions are made about the readers of this report: (1) they should be acquainted with the goals and objectives of the IIDA project, as presented in the 5 reports preceding this one[1]; (2) they should have some knowledge of the overall configuration of the IIDA computer system, although this will be briefly reviewed below; and (3) they should have had some exposure to the terminology used in computer and information science. Terms in the latter category will only be explained when they have some special meaning in the context of the IIDA system. Finally, in referring to users, the third person singular pronoun, "he", will be used. This should be understood as a reference to users of both genders.

From the broadest perspective, the IIDA system consists of four components: users at terminals, a computer with IIDA software, a bibliographic search service, and telecommunication networks which link users to IIDA and IIDA to the search service. The terminals happen to be common DECWriters, although any general-purpose ASCII terminal with an attached modem would work just as well. IIDA software resides on Massachusetts Institute of Technology's Honeywell 6180, a large-scale interactive timesharing machine running under the MULTICS operating system. The search service is provided by Lockheed's DIALOG system, an online information retrieval system which provides access to numerous bibliographic data bases. Either one of two commercial networks may be employed for the two links, specifically the Tymnet and Telenet services.

IIDA's software is programmed in the MULTICS extension of PL/1. M.I.T.'s MULTICS machine also provides a special hardware device, known as an Autocall, for placing outgoing calls to a network. IIDA software resides in seven directories, i.e., file and program storage and work areas, on MULTICS. These include: two entry directories for IIDA staff; one development directory for IIDA PL/1 source code; one development directory for IIDA object code; and

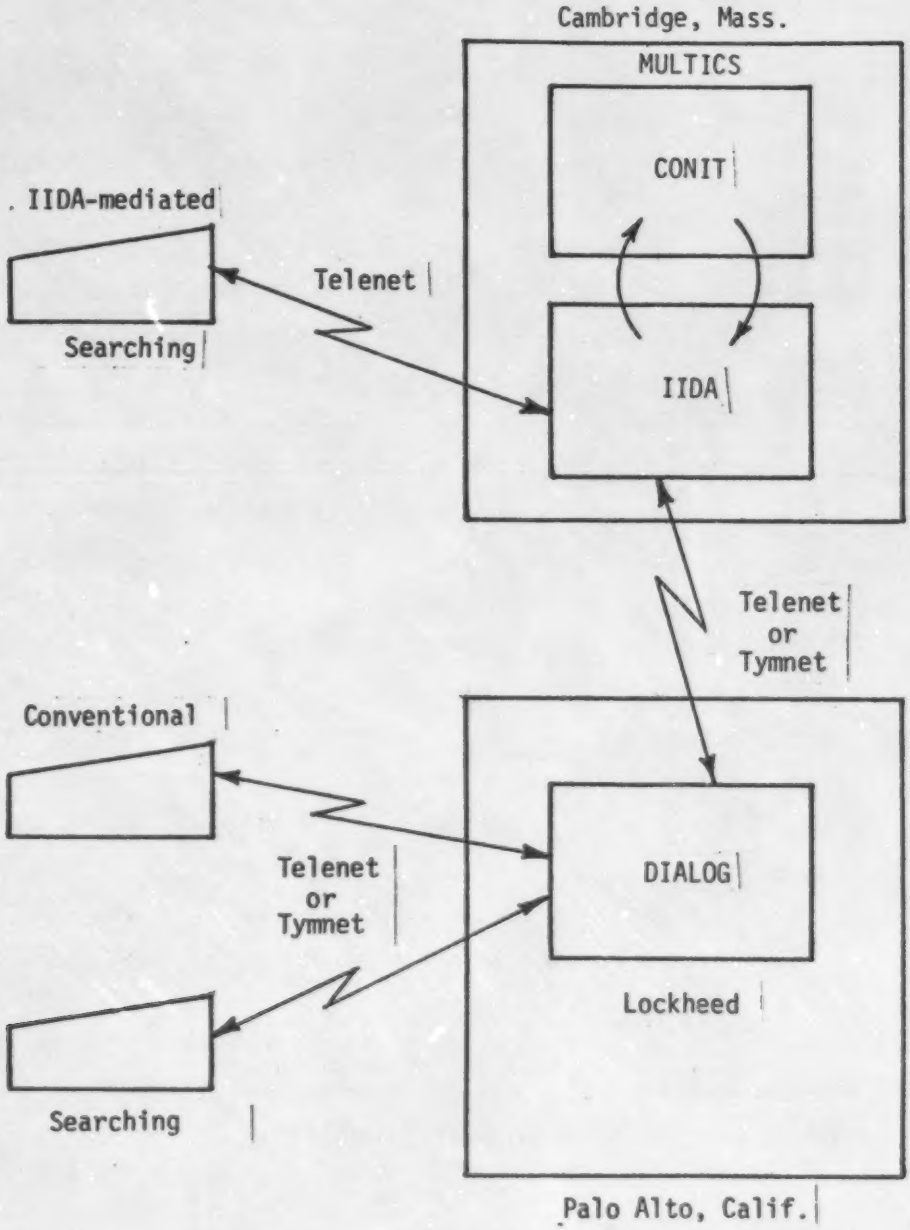

Figure 0. Computer Communications Configuration. Unmediated searching involves a direct connection to the host retrieval system. Mediated searching involves two connections. A connection is first established with the mediating software on an intermediate computer. A call is then placed by this software, via an AutoCall hardware device, to the host retrieval system.

three entry directories for guest users, i.e., experimental subjects, of the IIDA system. The directories are interlinked where necessary.

The software description in this report will be organized according to the programs that make up the system. Each program has a function sufficiently discrete to make this approach feasible. No discussion of IIDA software would be complete, however, without recognition and thanks to the Connector for Networked Information Transfer (CONIT) project[2] at M.I.T. from whom IIDA received most of its communication and control software. Chapter 2 of this report will introduce these essential programs which IIDA inherited.

See Figure 1 for a schematic classification of the system's programs. Chapter 3 of this report will describe the control programs. Chapter 4 will describe the search-oriented programs. Chapter 5 will describe the other support programs. See Figure 2 for the names and summary descriptions of the programs in this scheme. In the MULTICS system, all PL/1 source programs require a name-extension of ".pl1", pronounced "pea-el-one". This extension will be given throughout this report when referencing a program. This is to emphasize that the reference is to a program, although the executable program is in object form, the name of which takes no extension on the MULTICS system.

2. CONIT AND IIDA

The IIDA project was fortunate to be able to acquire and build upon system software developed by the CONIT project at M.I.T. The idea of a network interface is common to both projects, although the two projects differ significantly in the use they make of this interface capability. IIDA thus has made extensive use of the more general and system-oriented CONIT programs: those which perform control and communication functions.

The main program of each system performs the central control functions. This program came from CONIT named conit.pl1, and was renamed iida.pl1 for use by the IIDA project. Iida.pl1 manages the receipt and transmission of most messages to and from the user and to and from DIALOG. Message management and numerous other functions are performed by a production rule interpreter developed by the CONIT project. This interpreter is now part of iida.pl1.

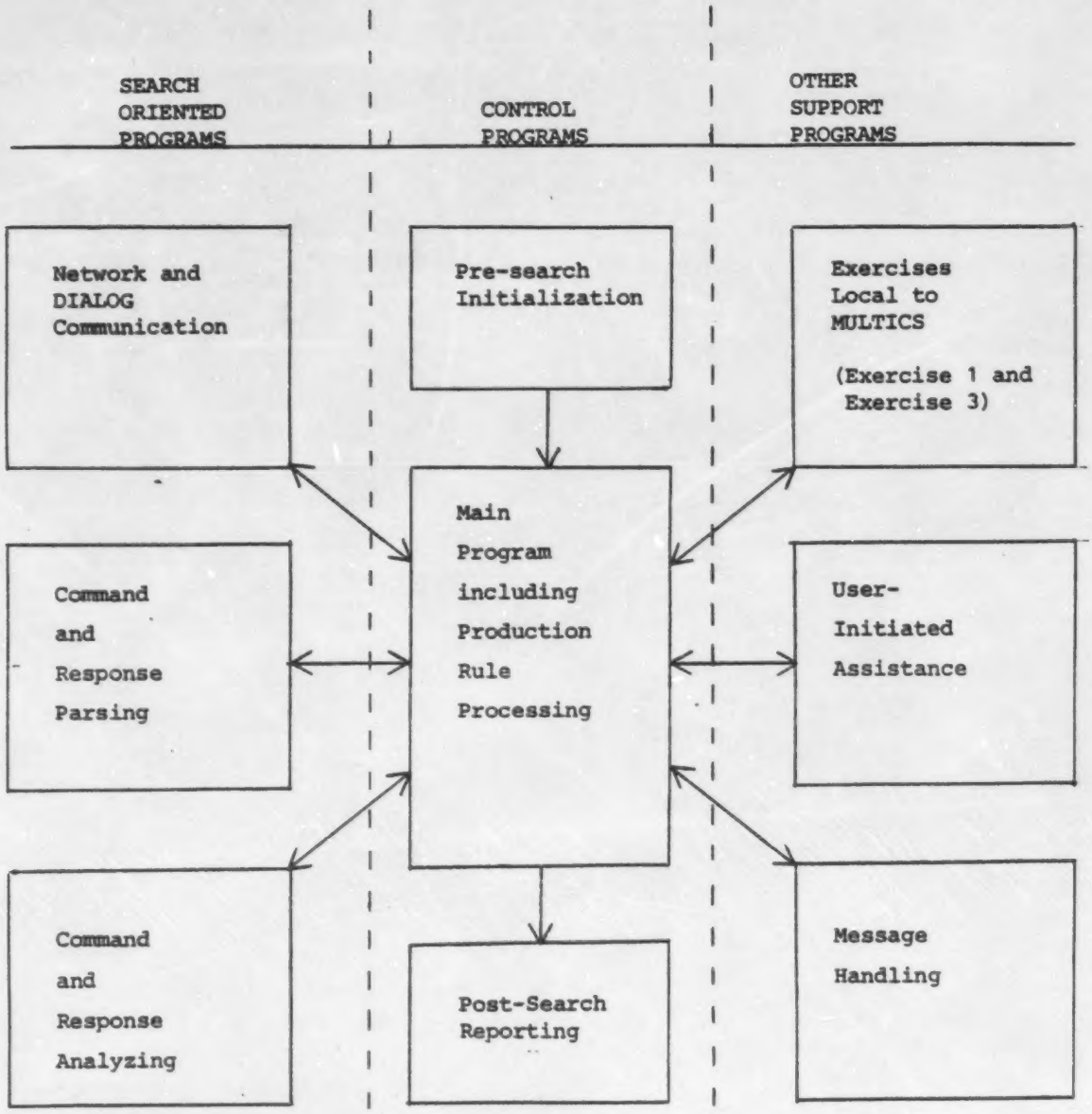

Figure 1. Schematic classification of programs available with the IIDA system. Procedures in the left column are immediately concerned with the particulars of messages transmitted between users and the search service in a given search. Procedures in the middle column are concerned with controlling the sequence of events during an IIDA search. Procedures in the right column serve informative and maintenance functions which in themselves are not sensitive to the particulars of the current search.

I. Control Programs

A. Pre-Search Initialization
   1. get_user_id.pll - get and verify identity of user
   2. db_init.pll - initialize selected variables in the user data base
B. Main Procedure
   1. iida.pll - control message switching and subsystem execution
C. Post-Search Reporting
   1. proc_rep.pll - online report of search activity for proctor or user
   2. reports.pll - store report of search activity for later printing
   3. adapt_rep.pll - online report of adaptive variable subsystem

II. Search-Oriented Programs

A. Network and DIALOG communication
   1. netcon.pll - connect to a network through an available i/o port
   2. callout.pll - place a phone call to the selected network
   3. sendline.pll - send a line of information to the search service
   4. getline.pll - get a line of information from the search service
B. Command and Response Parsing
   1. parse.pll - verify syntax of user-entered command
   2. rsvparse.pll - decode response line from the search service
   3. idverb.pll - analyze commands with unrecognized verbs
C. Command and Response Analysis
   1. canalysis.pll - analyze search strategy based on command elements
   2. normalizer.pll - standardize command arguments for subsequent analysis
   3. ranalysis.pll - analyze search strategy based on response elements

III. Other Support Programs

A. Exercises Local to NULTICS
   1. ex1.pll - manage Exercise 1 frames and responses
   2. ex3.pll - manage Exercise 1 and Help 1 frames and responses
B. User-Initiated Assistance
   1. help_iida.pll - offer user help from the data base and Ex. 3 frames
   2. da.pll - experimental help in getting useful descriptors
   3. rec_assist.pll - experimental help in getting useful records and desc.
C. Message Handlers
   1. messen.pll - general packaged messages for various system functions
   2. mesyner.pll - syntactic error message handler
   3. mess_out.pll - strategic condition message handler

Figure 2. Outline classification of program names and their summary descriptions. These programs are described in detail in the text. The above outline roughly corresponds to the table of contents of this report. Other minor support programs and data files are also discussed in the text, where it is fitting.

A production rule is an executable entity. It is selected from a linked list of rules, based on the current state of the process and data in an input stream. Execution of the rule establishes a new state, places data on output streams, and performs various special actions. Selection of production rules, one at a time, continues throughout the process — essentially producing or controlling the process.

The production rules effectually make up a sublanguage. A program in this sublanguage is stored on an external file, named ttstart. Iida.pl1 can invoke rul.pl1, a program which can add, delete, and change production rules on ttstart. The general features of the interpretive production-rule language are described elsewhere.[3,4] Iida.pl1 is discussed in more detail in Section 3.3.

Four programs which manage communication were received from CONIT and are used practically intact by IIDA. Two are concerned with establishing a communications link via the Autocall device to one of the two networks. These programs are named netcon.pl1 and callout.pl1. A third program, named sendline.pl1, is responsible for passing messages to the network and the search service over the established channel. A fourth program, named getline.pl1, is responsible for receiving messages from the network and the search service over that channel. Of these four programs, significant changes have only been made to getline.pl1 for the IIDA application. All four programs are discussed in greater depth in Section 4.1.

3. CONTROL PROCEDURES

3.1. Start-up.ec and search.ec

MULTICS has a facility for executing a list of monitor commands contained in a segment. This facility, called "execute commands", includes branching capabilities and may be applied to commands in segments with the name-extension ".ec". A segment with the reserved entry-name "start_up.ec" is automatically executed immediately after logging into MULTICS. IIDA's control programs and other preparatory activity is supervised by start_up.ec segments found in each of the three guest directories and by user-invoked executable command segments

named search.ec found in the two staff directories. The start_up.ec segments for guest users make logging into MULTICS and invoking IIDA appear to be a single operation.

The five executable command segments contain some variations, but generally follow these steps: (1) transfer control to the development directory containing the object segments; (2) accept or reject an IIDA-specific identification code; (3) initialize the data base; (4) invoke the main IIDA program; and (5) report on the search. Each of these steps deserves some amplification.

(1) Transfer control to the development directory. The executable command segments are found in the five entry directories (staff and guest). To execute IIDA software, either control must be transfered to the development directory where object-code segments for the IIDA system are stored, or links must be permanently established to all programs from the entry directory to the object-code development directory. In several instances the latter was done in order to isolate in the various entry directories data segments with the same name. This, for instance, gives unique pathnames to multiple copies of the student data base segments, found in several directories.

(2) Get an IIDA-specific identification code. Each IIDA user is assigned a unique code in each experimental environment. This is used to correlate search transcripts with reports printed later about the search. Some 200 codes are available. Some of the characters of the 9-character code serve switching as well as identification functions. Accepting the code is managed by the program, get_user_id.pl1. This program is found in the context of conditional branching in the executable command segment which externally handles accepted and rejected codes. Both the program and the codes are discussed in Section 3.2.1.

(3) Initialize the data base. The program, db_init.pl1, discussed in Section 3.2.2, initializes many of the external variables and structures found in the student data base, discussed in Section 3.2.3. An area (in the technical PL/1 sense) defined in the student data base must also be initialized by the MULTICS monitor command "create_area".

(4) Invoke the main IIDA program. The main IIDA program, iida.pll, is discussed in Section 3.3.1. It contains three auxillary entry points which provide either regular (guest) or terse (staff) entry into the IIDA software. The appropriate entry point is invoked by the executable command segment.

(5) Report on the search. Upon return from the main IIDA program, the user has the option of reviewing his search online. A copy of this search report is also appended to a data file for later printing and analysis. Both the online and filed reports are discussed in Section 3.4.

3.2. Pre-search initialization

3.2.1 get-user-id.pll. This program requests and verifies a user identification code -- a 9-character string which is used to correlate users with the reports printed later on their IIDA-mediated searches. Approximately 200 codes are stored in a table built into this program. Users are given three opportunities to enter a code found on this table. If a valid code is not entered by the third time, the IIDA session is terminated.

The first four characters and the last character of the code are randomly selected, serving strictly to uniquely identify the user. The code also includes switches that control program operation. Character 5 is a binary switch that indicates whether or not a report is to be printed later about the session. In practice, reports are not printed for staff users, but are printed for guest users. Character 6 is a binary switch which allows IIDA to be suppressed. This make IIDA activity transparent; an IIDA search appears to be exactly like a DIALOG search after logon is completed, although data is collected on the search (see Quarterly Report 4). This transparent or IIDA-suppressed mode is used by experienced control groups in the experimental studies. Characters 7 and 8 were intended to be binary switches that signalled the suppression of front-end material and the type of terminal. In practice, these two characters are not used as switches, but remain the constant, "lp".

3.2.2 db-init.pl1. This program initializes many of the variables in the student data base, discussed below in Section 3.2.3. The executable portion of db_init.pl1 are all assignment statements. Individual variables, arrays, and even entire structures, are assigned initial values. The selection of variables to be initialized and the sequence of their initialization was made in an ad hoc fashion. Key variables, such as indexes to arrays, were initialized to 0 from the beginning of software development. Other variables were added to this program as it became evident that they needed initialization.

3.2.3 student.incl.pl1. This segment contains the source of all declarations that make up the student data base. These variables are organized into six level-01 external static data structures. These data structures roughly represent: command history, other fixed-structure histories, variable-structure histories, data on EXPAND tables, supporting data, and data for the adaptive-variable subsystem.

The command history contains variables parsed out of or analytically derived from command strings as entered by the user. This data includes: argument of the command, various classifications of the command, set generated (if SELECT or COMBINE), time taken to enter the command, and various condition signals.

Other fixed-structure histories are organized into arrays that represent information returned by DIALOG. These include: sets created by SELECT and COMBINE, records reviewed by TYPE and DISPLAY, use made of the IIDA HELP facility, syntactic errors identified by the IIDA parser, and strategic errors identified by the IIDA analyses. Entries in each of these structures contain pointers to the entry in the command history structure that triggered the retrieval system response.

Variable structure histories consist of two PL/1 based structures allocated in a PL/1 area. These structures contain: 1) extended arguments of set-creating commands; and 2) a history of all descriptors used either explicitly or implicitly in the search. The derivation of extended arguments is discussed in Section 4.3.2.

Data on EXPAND tables includes the entire parsed image of the most recent EXPAND table and the most recent thesaurus table (if any). Also stored here are the first and last terms from every response in the search to a lexical (not thesaural) EXPAND command. This last table is used in the analysis of null sets (see Section 4.3.3).

Support data contains by far the largest collection of simple variables. Stored here are: the parsed elements of the user identication code; indexes to all arrays contained in other segments; and data generated in analyses of search commands, DIALOG responses and the search as a whole. Various flags and summary data are also found here. The use of many of these variables is discussed in Section 4.3, strategy analysis.

The adaptive variable subsystem is discussed in Section 4.3.3. Data for this subsystem is stored in a distinct external structure. The structure consists of 21 arrays whose values are initialize by db_init.pl1. These values are modified when conditions are identified by the strategic analyses.

An image of the entire student.incl.pl1 segment has been printed in a previous report[5]. The variables, as named and declared there, have not changed significantly. The student.incl.pl1 source code was included in the compilation of most of the IIDA programs. This allowed each to access and update external variables with values used by the other programs of the system.

Each of the six data structures has essentially been compiled into an object segment. These allow data maintained in the student data base to be accessed while the session is underway and after the session is through. The original intent of having permanent external data was to allow an IIDA staff member, known as the proctor, to review the searcher's progress from a separate terminal during the search. Although this capability exists with the proc_rep.pl1 program (discussed in Section 3.4.1), it generally has not been done in practice. With the data externally available, however, online reporting to the user and file storage of the search are invoked at the command processor level (search.ec) rather than by a call by the main program (iida.pl1).

3.3 Main Procedure

The main IIDA procedure is iida.pl1. This procedure centrally calls for all activity: introducing IIDA, handling the four exercises, managing logons to DIALOG, managing all search communications, and logging off of IIDA. Controlling the apparent flow of all this activity are CONIT-style rules stored in the compiled segment, ttstart. The functions of iida.pl1 will be discussed in Section 3.3.1; the flow defined by ttstart will be discussed in Section 3.3.2.

3.3.1. iida.pl1. The main PL/1 procedure, iida.pl1, performs a number of functions. These include: providing 3 alternate entry points, initializations pertinent to processing production rules, handling interrupts, handling messages to and from the retrieval system and to and from the user, and interpreting the production rules. Each of these functions will be discussed in turn.

Iida.pl1 may be entered by a regular connection (rcon), a terse connection (tcon), or an IIDA staff connection (icon). Guests use the entry point rcon, while IIDA staff uses the other two entry points. Entry at rcon presumes that data base initialization and verification of user identification have already done by db_init.pl1 and get_user_id.pl1 under the control of start_up.ec. The staff entry points essentially perform these two functions. Entry at tcon gives the staff user an immediate prompt, while entry at icon gives the staff user the same preliminary material seen by guest users.

Some initialization is done at the beginning of iida.pl1. This initialization pertains almost entirely to the program's subsystem that interprets the production rules. In particular, ttstart, the compiled form of the production rule table, is attached to the process. Another production rule table, should there be one, may replace this under control of certain production rules. Only ttstart is used in practice. Switches, flags and other system-software variables are initialized here. This is in contrast to the IIDA application-software variables initialized by db_init.pl1.

Two interrupt conditions are handled by iida.pll. These are the user-signalled QUIT and the system-signalled phone hangup. The QUIT condition is raised when the user strikes the BREAK key. The hangup condition is raised either when the user indicates he wants to be dropped from the host (LOGOFF) or when the phone connection is accidentally broken. Raising either condition results in changing a variable representing the current context of the search. Translation begins on a whole alternate set of production rules written to handle these conditions.

Not all messages sent to or received from the user are managed from iida.pll; some of the called programs perform both functions. Commands directed to DIALOG or IIDA in response to the "?" cue and most initial responses to such command are controlled by iida.pll, however. Almost all messages sent to and received from the retrieval system are controlled here.

Statements immediately following the label "luser" (listen to user) manage user responses to the cue. This cue may be "?" for IIDA-suppressed usage, "D?" for commands to be directed to DIALOG or IIDA, or "I?" for responses to questions from IIDA. The user's entry is timed and recorded. If the user takes more than 90 seconds to make an entry, he receives a warning; if more than 3 minutes, he is logged off. Stacked commands directed to DIALOG are split up into separate commands for later analysis, and the individual strings are prepared for processing by the production rule table. This normally includes the conversion of all input from upper case to lower case.

Statements immediately following the label "lrs" (listen to retrieval system) manage DIALOG responses to commands. One line of response at a time is handled through a call to getline.pll, discussed in Section 4.1.4. Tests are made for various timeout codes returned by getline.pll. If these conditions, signalled by codes, are present, the current context string is altered, triggering the interpretation of sets of production rules handling these conditions. The line(s) returned are prepared for processing by rules in the table.

The rule table translator/interpreter functions as a complete algorithm within iida.pll. This algorithm is explained in more detail elsewhere[6], but a summary description is in order: At all times, a variable named "pref"

maintains a string of codes reflecting the current state of the process. The rule interpreter searches the table for a rule whose state-map matches "pref", either completely or partially. Rules also contain match-strings for designated input streams. Among all rules with a state-map matching "pref", the rule with the longest matching match-string is selected.

When this "best rule" is found, messages may be constructed for either the user, the host or both. The rule also indicates changes to "pref", reflecting a new state entered by the process. Any one of 70 special action may be executed before the rule interpreter seeks the next rule. These special actions include some which are system-oriented (received from CONIT), and some which are application-oriented (developed by IIDA). Most of these special actions involve calls for the execution of other IIDA programs. In the following discussion of special actions, called programs will be indicated in parentheses.

System-oriented special-actions can be reduced to roughly the following categories: maintain and display rules in the production rule table (rul.pl1); place and terminate phone calls to networks (netcon.pl1, callout.pl1); set timing parameters -- tolerated and generated delays; numerous variants of setting up and sending messages to users (messen.pl1); numerous variants of setting up and sending messages to the retrieval system (sendline.pl1, netcon.pl1); elementary numeric activities -- store, place, add, and compare numeric values; and return to the MULTICS command level.

Application-oriented special-actions can be reduced to roughly the following categories: call for parsing of DIALOG-directed commands (parse.pl1); call for the strategic analysis of these commands (canalysis.pl1); call for parsing of DIALOG responses (rsvparse.pl1); call for the strategic analysis of these responses and handling of messages triggered by both analyses (ranalysis.pl1); manages DIALOG-directed commands entered from external procedures as returns from those procedures (help_iida.pl1, da.pl1, rec_assist.pl1); manage the BEGIN command issued in the middle of Exercise 4 (db_init.pl1, reports.pl1); manage all IIDA-directed commands (help_iida.pl1, da.pl1, rec_assist.pl1); handle several classes of exceptional errors (idverb.pl1); and provide for simulation of retrieval system responses locally on MULTICS.

3.3.2 ttstart. This segment is the "compiled" version of the production rule table used throughout the IIDA session. Compilation is performed by a call to the program rul.pl1. Following any IIDA prompt, "?", rules may be entered as strings prefixed by the command "rul". This invokes rul.pl1 in which the strings are compiled into ttstart as a list of structures containing the components of each rule. Images of these strings are also maintained in an ASCII segment named iida_erul. The apparent flow-of-control of the entire process is managed by these rules. See Figure 3 for a general schematic of this flow-of-control. Following is a discussion of this flow, as controlled by groups of production rules.

The first rules interpreted initialize selected variables in an array which the production rules can access and test. These initial values depend in some cases upon the entry point used. Among these variables are: user id, intended use of IIDA, switch and flag initializations, etc. If IIDA is suppressed, that is, the user id was issued to a profession search intermediary in the control group, the next rules interpreted are those that control placing the outgoing phone call and login protocols. Otherwise, the user is assumed to be a novice and several introductory activities take place.

These activities include an opportunity for the novice searcher to read an introduction to the IIDA concept and system. This is broken down into several frames. After any frame the user may decide he has seen enough introduction. When through with this introduction, the user is asked to indicate the mode with which he wants to work. These include the three exercises and the Assistance Mode. If Exercise 1 or 3 is chosen, rule interpretation is temporarily suspended after a special action calls the procedure ex1.pl1 or ex3.pl1 to manage the chosen exercise (See Section 5.1). After returning from these exercises, the user again is given the initial menu from which he many choose a mode.

If Exercise 2 or the Assistance Mode is chosen, the user is asked, under rule control, for the approximate number of records he hopes to retrieve. This is followed by messages reflecting the interpretation of rules that place a phone call (see Section 4.1) to the Tymnet node and manage login protocols

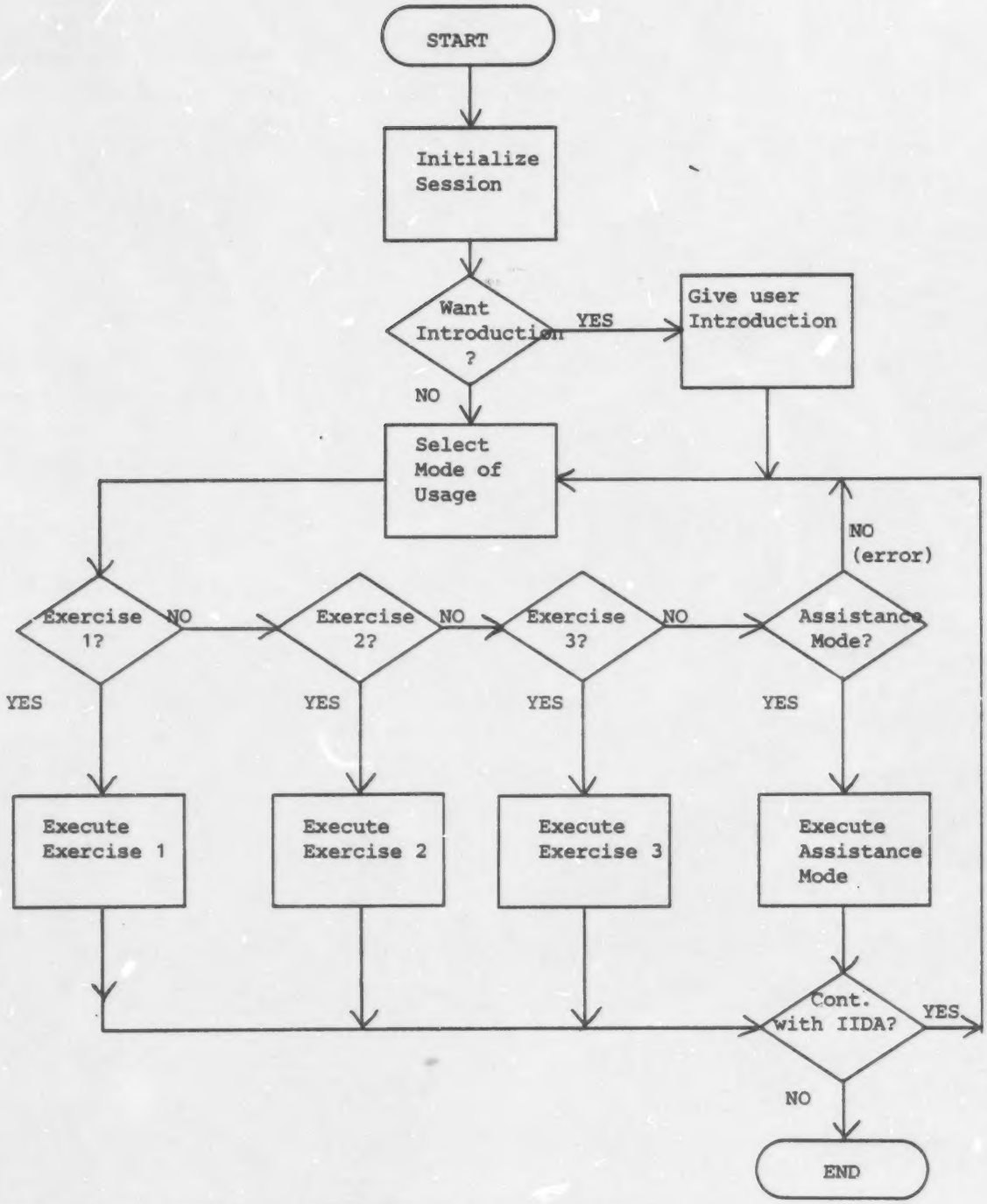

Figure 3. Overall flow-of-control of the IIDA system. See Figure 4 for an overview of Exercise 1 and Exercise 3 flow-of-control. See Figure 5 for an overview of Exercise 2 and Assistance Mode flow-of-control.

to DIALOG. If a failure occurs at this point, the phone is hung up, a phone call is placed to the Telenet node, and Telenet's login protocols for DIALOG are executed.

Once online to DIALOG, the user receives a prompt for the entry of either DIALOG-directed or IIDA-directed commands. The verbs of all legal commands are found in the match string of a set of rules. If no match is found, an analysis is made of the input string by idverb.pl1 (see Section 4.2.3). If a match is found, the command parser is invoked through a special action (see Section 4.2.1). If the command contains an error, this is explicitly described for the user (see Section 5.3.2) and he again receives the cue. Otherwise, the message is sent to DIALOG and the command-dependent strategy analyzer is invoked under rule control (see Section 4.3.1). For each line returned by DIALOG, the response parser is invoked (See Section 4.2.2). When the current context indicates the end of transmission from DIALOG (see Section 4.1.4), control passes to the response-dependent strategy analyzer (see Section 4.3.3). Occassionally, the response is an error message from either the network or the retrieval system. In this latter case, the message is amplified by IIDA. Finally, IIDA returns another input cue to the user.

In recognizing DIALOG-directed commands, the production rules handle several exceptions. For instance, they recognize but do not transmit the command verb PRINT. Rather, an error message is returned explaining that although PRINT is allowed by DIALOG, it is not allowed with the IIDA experiment. Similarly, in Exercise 2, BEGIN may be issued only once, at the beginning of the search. BEGIN may be issued at any time in the Assistance Mode, where it triggers rules that call for a report to be stored on the file and the student data base to be reinitialized. When TYPE is recognized, an explanation of relevance rating is first displayed (see Section 4.2.2). When LOGOFF is recognized, several operations and special actions are initiated to wind up the session.

Recognition of an IIDA-directed command triggers the interpretation and execution of rules followed by special actions which may invoke external procedures. Among these commands are: Quick Advice, SLACK, Descriptor Assistance, HELP, and DONE. Quick Advice prints advisory messages keyed to the codes of strategic conditions signalled and described to the user. SLACK

can be used to reduce the frequency of strategic messages controlled by the adaptive variable subsystem. Descriptor Assistance is controlled by the interplay of production rules and an external procedure (see Section 5.2.3 and 5.2.4). This provides special assistance in locating descriptors that may be useful in the search.

The IIDA-directed command HELP is also controlled by the interplay of production rules and an external procedure (see Section 5.2.1). The command may take a numeric argument. If HELP is not given an argument, an initial menu of various types of assistance is given the user. If a whole number argument is given, the table is bypassed, and the assistance, usually a summary of the search, is immediately given. If the number is a decimal fraction whose value ranges between 1 and 2, the fractional part is converted into a whole number which is then treated as the number of an instructional frame accessed and printed by ex3.pl1 (see Section 5.2.2).

The IIDA-directed command DONE allows the user to quit an exercise or mode, then either quit the session or begin another exercise or mode. There also exists a substantial set of IIDA-directed commands that are reserved for use by the IIDA staff. The first character of many of these must be the period. The command "rul" belong to this class of commands, as does ".lt" (list segment of ttstart), ".test2" (simulate DIALOG responses to an Exercise 2 session), ".stop" and ".det" (quickly terminate session), ".setlc" and ".setuc" (switch on and off conversion of input alphabetic characters to lowercase), etc.

There are also several interrupts handled under rule control. The quit (BREAK) and the phone hangup interrupts were discussed in Section 3.3.1. Code in the main program works in conjunction with production rules to handle them. Interrupts in receiving response lines from the retrieval system are also recognize by getline.pl1 (see Section 4.1.4), and inserted into the current state string, "pref", by the main program's rule processing algorithm. Alternate paths in interpreting production rules follow the new state established by these interrupts.

3.4 Post-Search Reporting

When the main program returns to the command level, the process is still under the control of search.ec (staff-invoked executable command segment — see Section 3.1) or start_up.ec (guest-invoked executable command segment). The six object versions of the student data base contain values from the last search and remain available to the process. Three programs in the IIDA system are able to report on this data. These are help_iida.pl1 (Section 5.2.1) which reports to the user while the search is underway; proc_rep.pl1 (Section 3.4.1) which reports on the search at the terminal; and reports.pl1 (Section 3.4.2) which writes the same report about the search to an extendable file. Finally, adapt_rep.pl1 (Section 3.4.3) reports on the adaptive variable subsystem at the terminal, but must be explicitly invoked by the user from the command level.

3.4.1 proc-rep.pl1. This program generates a report at the terminal about the search of either the current user or the last user of the IIDA system. A proctor at one terminal can invoke this program to review the progress being made by a user at another terminal. Alternately, a user can get a report on his search after he has completed it. The user may even log off MULTICS, log on again, and get the report, since all values stored for the report are in the permanent external segments of the student data base.

The report is organized into seven tables. These describe the search in terms of histories that roughly correlate to the structures in the student data base. All but two of these tables are organized chrononologically. These tables report on: command history; sets created history; records viewed history; chronological history of all possible errors; classified history of strategic conditions; descriptors used history; and help invoked history. The detailed contents of each of these tables will now be discussed. See Appendix 1 for a sample report.

Command history reports on every command entered by the user, including those directed to both DIALOG and IIDA. This table consists of nine columns: (1) sequential numbering of commands; (2) class code indicating DIALOG-directed, IIDA-directed, or syntax error; (3) command classification

(see Figure 5 in Quarterly Report 2); (4) set created, if command was SELECT or COMBINE; (5) string classification within group; (6) group classification within search; (7) stack association, if any; (8) command verb spelled out in full; and (9) argument of command. (String and cycle are defined on page 14 of Quarterly Report 2.)

Sets created history reports on every set created by the user during the course of the search. This table consists of seven columns: (1) command numbers corresponding to numbers in column 1 of the command history table; (2) sequential numbers of sets created; (3) postings count of sets created; (4) number of times set is subsequently referenced by COMBINE and TYPE commands; (5) average relevance, if records from set were viewed with the TYPE command; (6) full command verb; and (7) extended argument in which descriptors are spelled out in full.

Records viewed history reports on all uses of the TYPE command. This table consists of seven columns: (1) command numbers corresponding to the numbers in column 1 of the command history tables; (2) set viewed: the first element of the TYPE argument; (3) format: the second element of the TYPE argument; (4) first record: the first record number of a range; (5) last record: the second record number of a range, if any; (6) average relevance: average of all relevance scores assigned by the user to records viewed with the current command; and (7) argument of the TYPE command.

Chronological history of all possible errors reports on syntactic errors, and strategic and hybrid conditions. Syntactic errors are identified by the command parser conditions and are not transmitted to DIALOG. Strategic conditions are identified by the analysis of commands and responses, individually and in the context of the search as a whole. Commands with these conditions are transmitted to DIALOG. Hybrid conditions are IIDA restrictions that are essentially strategic in nature, but are identified by the syntax parser and are generally not transmitted to DIALOG. This table consists of five columns: (1) command number corresponding to the numbers in column 1 of the command history table; (2) error code: a summary list of syntactic error codes may be found in Appendix 3 of this report, while a definitive list of strategic condition codes may be found Appendix 4; (3) error type: syntax, strategy, or hybrid; (4) command verb; and (5) command argument.

The classified history of strategic conditions organizes each strategic condition into one of 10 classes. The names of these classes and their corresponding codes are given in Figure 4. The table printed has five columns: (1) condition class code (1-10); (2) detailed condition code (1-45); (3) number of times the condition was signalled; (4) the command number of the last command in which the condition was signalled; and (5) the class description. The table is organized by class code. Each time this code changes, a line summarizing the number of conditions identified in the class is printed.

The descriptors history is ordered alphabetically by descriptor used. All descriptors and terms in the search are cataloged here. Indirect references to terms are made explicit. For instance, selections from an EXPAND table are converted into the corresponding descriptors from the table. The descriptors history table consists of five columns: The first four contain the number of times a descriptor is referenced in the argument of the four commands: SELECT, EXPAND, thesaural EXPAND, and COMBINE. The fifth column contains the descriptor itself.

The help-invoked history table is again chronological. As it currently operates, the table is somewhat obsolete: no argument is listed although the help command can take literally hundreds of arguments because frames of the Exercise 3 file can be directly referenced. Four column are printed: (1) time help was requested (from a 24-hour system clock); (2) time help request was terminated with the user returning to the search; (3) minutes and seconds spent in help; and (4) type of help requested.

3.4.2. reports.pll. The function of this program is to store a report on every search. The stored report looks almost exactly like the one printed at the terminal when proc_rep.pll is invoked. Instead of printing immediately, however, the report is stored on an external, extendable file. This file is periodically printed on a high-speed printer at M.I.T., mailed to the IIDA staff, and deleted from disk storage. Date, time, and user id code are printed as a heading to every report. Since a report is written following every experimental search session, a complete chronological history of the use of Exercise 2 and the Assistance Mode can be reconstructed. If the id code is that of an IIDA staff member, no report is written.

Class Code	Strategic Condition Class Name	Strategic Condition Class Member Codes
1.	String/cycle length	3, 4, 5, 6, 7, 8, 39
2.	Null set generation	9, 10, 11, 12, 13, 14, 36, 44, 45
3.	Uninformative formats	18, 30
4.	Non-used sets	19
5.	Command Repetition	2, 22, 23, 38, 40
6.	Thrashing	24
7.	Dwelling	25, 31, 32, 33
8.	Relevance Comments	26, 27, 28, 29, 30, 41, 42, 43
9.	Time to enter command	34
10.	Viewing Request Excessive	35

Figure 4. Strategic condition classes and their corresponding codes. These codes are found in the Strategic Error proctor report, a sample of which is found in Appendix 1. The detailed description of the condition, by code, is found in Appendix 4.

This program saves the seven tables corresponding to the proctor report, as well as an eighth table corresponding to the adaptive variable report. That report is explained in detail in Section 3.4.3. For one experiment performed in an industrial R&D environment, all descriptors that appear in arguments of commands and in the descriptors history table have been converted to dummy terms. This preserves the propriatary nature of the search subject by masking the intellectual content of the search, while maintaining intact the form of the search.

3.4.3. adapt-rep.pll. This program reports on the adaptive variable subsystem, printing at the terminal a table reporting on this subsystem. A brief description of the adaptive variable subsystem is in order before the columns of the table are discussed. A typical table returned by adapt_rep.pll is reproduced in Appendix 2.

The adaptive variable subsystem is a technique for selecting messages associated with strategic conditions signalled by the strategic analyses. Forty-five different strategic condition may be signalled. (In previous IIDA reports, "conditions" are called "procedural rules" and "strategic errors"; the three terms are essentially synonymous.) Several different conditions may be raised with each command entered by the user. It is possible for the same condition to be raised for several commands is a sequence of commands. This is the extreme case in which we do not want to blindly call for the same message every time, dunning the user.

Messages that would appear too frequently are suppressed by both algorithms and a schedule built into the Warning Control Program (see Section 4.3.3). Instances of algorithmic suppression are: generic messages are suppressed when signalled with specific messages with the same intent; if a message was sent less than three commands ago and the same condition is raised again, the message is suppressed. This last instance exemplifies non-adaptive linear message suppression. Some message suppression is non-linear; these are selected by the adaptive variable subsystem.

Conditions for which adaptive non-linear message suppression applies are mapped into 15 sets of adaptive variables. Each variable has its own

distinctive message scheduling curve, which generally increases the frequency of messages, and which may be reset to the initial frequency at string or cycle breaks. Message frequency, in other words, is a function of the number of times conditions are signalled and the state of the search in terms of its string and cycle structure.

This trend of increasing the frequency of messages can be relaxed by user request. Following the receipt of any strategic condition message, the user may enter the IIDA-directed command /SLACK. This effectually decreases the message frequency by increasing the current interval between approving messages for the currently raised conditions.

The adaptive variable report consists of six columns. There are 15 rows to the report. Each row represents one of the adaptive variables. The sixth column sequentially numbers these rows with the variable index. The first column classifies the variables into the following six classes (variable indexes in parens): essential command repetition (1,2); string length (3,4,5); null sets (6,7,8); non-used sets (9,10); exact command repetition (11,12,13); and dual command repetition (14,15). Column 5 counts the total number of adaptations for the given variable. Columns 3 and 4 indicate the variables' initial and final values. These values represent thresholds tested in the strategic analyses. If the thresholds are exceeded, the condition is signalled and the threshold is raised. Column 2 gives the difference between the initial and final values. The detailed meaning of the adaptive variable thresholds is given in Sections 4.3.1 and 4.3.3, and in previous IIDA reports.

4. SEARCH-ORIENTED PROGRAMS

This section is concerned with a set of 10 programs, called by iida.pl1, which primarily deal with the particulars of the messages exchanged by the user and the host retrieval system. Four of these are "system programs" involved in establishing a link to the host and transmitting messages to and from the host. Three of these programs have been used practically intact as received from CONIT. The fourth program has been adapted by the IIDA project.

The remaining six programs are "application programs" which evaluate the syntax of the messages exchanged (the parsers) and the strategic value of the messages in the context of the search as a whole (the analyzers). These programs have been written expressly for IIDA by the project staff. The flow of the usage of these programs is sketched in Figure 5. The system programs will be discussed in Section 4.1; the parsers will be discussed in Section 4.2; the analyzers will be discussed in Section 4.3.

4.1 Network and DIALOG Communication

4.1.1. netcon.pll. This program establishes and maintains the external link to the network. The main entry point of the program is called as a function whose parameters are: a network code, a channel code, and an error code. The value returned by netcon.pll is a supplementary error code. The network code may be either 1 for Tymnet, or 2 for Telenet; the channel code may be either 1 or 2 for two distinct 300-baud ports. The error codes are examined by iida.pll after a return from netcon.pll. Both are used in constructing an error message by the production rules, if necessary.

Netcon.pll test the availability of the requested channel. If it is not available, an attempt is made to switch to another channel. If no channel is available, a message is sent to the user informing him about the current usage of the telephone, if that information is available. If a channel is available, a call on that channel is attempted by calling the program callout.pll (discussed in Section 4.1.2). Information on the current usage is stored, the i/o stream is attached and opened, and i/o modes are set. Failure at any of these points causes a return to the main program with appropriate error codes.

Four minor entry points are available in netcon.pll. These allow for various i/o activities on the phone channel. These neither take parameters nor return values, but allow for changing i/o modes, resetting the channel, and closing and detaching the channel.

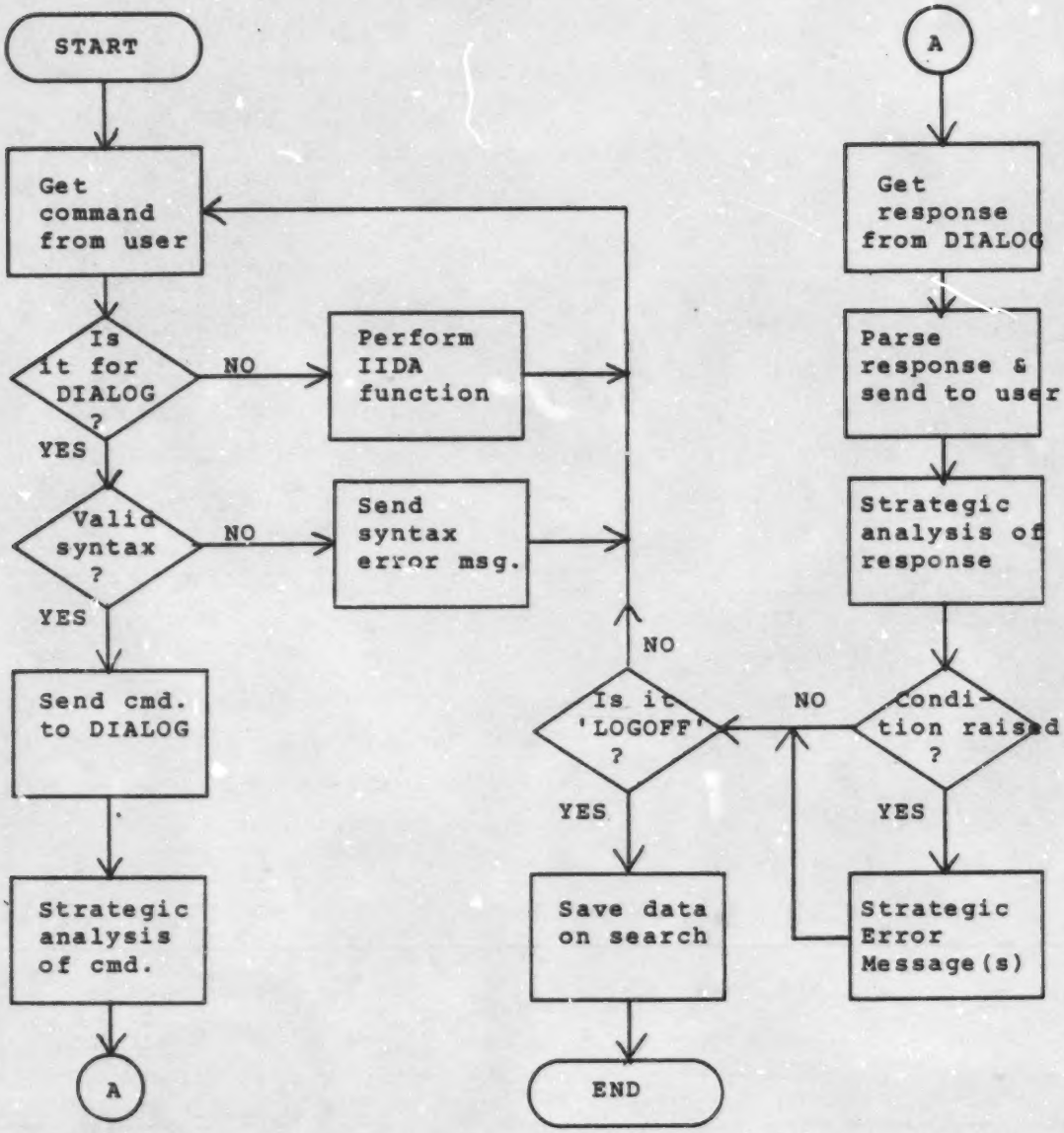

Figure 5. Overview of flow-of-control for Exercise 2 and the Assistance Mode. Exercise 2 is not as general as Assitance Mode in that Exericse 2: (1) requests in an introductory paragraph that one of two specific subjects be searched, and (2) allows only one BEGIN command to be issued during the session. The Assistance Mode has neither of these restrictions.

4.1.2. callout.pl1. This program controls the actual dialing of the phone number of the local network node. The main entry point of the program is called as a procedure whose parameters are: phone number of node, channel number, device (channel name), i/o module, and error code. The phone number is passed to callout.pl1 by netcon.pl1, while all the remaining parameters are given values in callout.pl1 to be returned to netcon.pl1. MULTICS system subroutines are invoked to create the event and place the phone call. Date and time are printed at the terminal before and after the critical subroutine calls, providing a window on the timing of this activity.

Two other entry points are available in callout.pl1. The entry point "hangup" is invoked by the operating system's interprocess communication overseer when the channel hangs up. This allows some cleanup activities to be performed on the channel. The entry point "endcall" is invoked to terminate an Autocall connection. This is called during the normal detach and drop activity.

4.1.3. sendline.pl1. This program controls the transmission of commands to the network or DIALOG over the channel previously established. The program's only entry point is called as a procedure whose parameters are: channel name, message, and error code. The channel is again that established by netcon.pl1. The message is any command directed by the main program to the network or DIALOG. The error code passed back to the main program indicates whether or not the message transmission was successful.

4.1.4. getline.pl1. This program controls the reception of responses from the network or DIALOG over the channel previously established. The program has only one significant entry point, invoked as a procedure with the parameters: channel name, maximum length of message, error code, and i/o mode flag. The response line received is passed back to the calling procedure, the main program, through an external variable known by both programs. The channel name is again that established by netcon.pl1. The error code returned to the main program indicates whether or not the reception was successful. The other two variables are no longer significant in the IIDA version of getline.pl1.

CONIT provided for both line and character i/o modes, indicated by getline.pl1's fourth parameter. Character input is particularly important for communication with DIALOG since the end of any transmission can only be definitively recognized by the input cue "?", which is a partial line. CONIT communicated with various hosts, while IIDA communicates only with DIALOG. The program getline.pl1, was thus revised to only receive messages character-by-character. This allows a quick and definitive recognition of the end of a DIALOG transmission.

A timed interrupt is an important part of this program. If no initial response line is received after 20 seconds or no subsequent response line is received after 5 seconds, the user is sent a message to this effect. After another delay, if no response is received, the program quits looking for a response and returns control to the main program where the current state string is altered to indicate this failure. If any characters appear in the buffer after the second delay, these are appended to the response string before control returns to the main program.

4.2. Command and Response Parsing

4.2.1. parse.pl1. This program has two major functions: (1) it evaluates whether or not DIALOG commands entered by the user conform to the syntax accepted by DIALOG; and (2) it isolates all significant elements of valid DIALOG-directed commands, storing them in the student history data base. The general form of the program is a function, called by a special action in the main program.

The parameters passed to parse.pl1 are a code indicating the command verb and the command argument. The value returned by parse.pl1 is a condition code ranging from 0 to 110. If the command is valid for transmission to DIALOG, parse.pl1 returns a value of 0 and the command is sent to DIALOG. If syntactic errors are identified, parse.pl1 returns a code between 1 through 99. If hybrid conditions are identified, parse.pl1 returns a code between 100 and 110. The command is not transmitted to DIALOG when a non-zero code is returned, with the exception of three codes which represent hybrid conditions.

Parse.pll consists of 12 sections, corresponding to acceptable DIALOG command verbs. These include: COMBINE, SELECT, EXPAND, TYPE-PRINT-DISPLAY, LIMIT-LIMITALL, BEGIN, EXPLAIN, DISPLAY SETS, PAGE, END, LOGOFF, and .FILE. Depending upon the verb code, one and only one of these sections is executed after parse.pll is invoked. Each of these sections consists of initialization, detailed analysis and windup activity. The detailed analysis of the six most significant sections will now be discussed.

(1) COMBINE. The parser first determines if the "compressed" form of the COMBINE command, e.g., COMBINE 1-4/OR, was entered. This is done by scanning for a slash (/) symbol. If the argument is of this form, the syntax is verified, the initial and final sets are isolated, and the extended argument in terms of the original descriptors is constructed. Syntax verification includes validating the general form, (first set)-(last set)/(operator), and determining if the first set exceeds the last set, if the last set exceeds the search's highest set, and if the operator is either "AND" or "OR".

If the argument is not of the compressed form, the boolean form is assumed. A character-by-character scan of the entire argument is performed. Parenthetical depth and state-transition values are maintained in the validation of the expression's syntax. The extended argument is constructed by translating each set reference in the argument into that set's corresponding extended argument. If a null set is used within an ANDed term, this is marked by a flag so that a warning can be sent to the user later. If the entire argument can be predicted to result in a null set, this also is flagged as a syntax error.

(2) SELECT. The parser first determines if a super-SELECT was entered by scanning the argument for the occurrence of logical operators. The present version of IIDA allows the Super-SELECT syntax only with the IIDA-suppressed option. Otherwise, an error code is returned. The parser otherwise scans the argument to determine if it: (a) contains a prefix, e.g., AU=; (b) contains a suffix, e.g., /TI; (c) refers to the last EXPAND table(s), e.g., SELECT E17; (d) contains infixes, e.g. SELECT COAL(W)TAR; or (e) is a full text argument.

Validation appropriate for each of these forms is performed. For example, the parser determines if the prefix and the suffix are valid for the current data base, it validates the syntax of infixes, and prohibits truncation to be done in an argument containing an infix. If the argument refers to EXPAND table entries, the parser makes sure that the references are within the limits defined by the last EXPAND table. If so, the argument is extended in terms of its member EXPAND table entries. This extension is stored in the student data base.

(3) EXPAND. The parser scans the argument to determine if it: (a) has a prefix; (b) refers to a previous EXPAND table entry (a thesaural EXPAND); or (c) if it is a direct reference to a thesaurus table entry (argument is parentherized). If it has a prefix, this prefix is validated for the current data base. If it refers to a previous EXPAND table, the existence of such a referent is validated. If the argument is parenthesized, matching of parentheses is verified.

(4) TYPE and DISPLAY. Elements of the argument are verified, isolated and stored in the student history data base. If any elements are missing, implied DIALOG default values are stored. If the argument refers directly to accession numbers, the format code, if present, is verified and saved in the student data base. If a suffix qualifier is present, the parser determines if it is allowed for the current data base. Otherwise, if the argument is not a reference to accession numbers, the set, format, and record-range elements are verified, isolated, and saved.

(5) LIMIT and LIMITALL. If a set is specified, it is verified, isolated, and saved. If it is limited to accession numbers, their general format is validated. That is, 6 digits must be present in each number and the final number must not be less than the initial number. If the argument is qualified, the parser determines if the qualification is valid for the current data base.

(6) BEGIN. If the command has an argument at all, it must one of the four data base numbers processed by IIDA: 1,6,8,201.

(7) All other commands. Validation is trivial. The argument must either literally match possible arguments in a list, or it must not present at all.

4.2.2. rsvparse.pll. This program has two major functions: (1) it determines if a response is normal and expected or if it is an error message from either DIALOG or the network; and (2) if the response is normal, it isolates the elements of the response for storage in the student history data base. The general form of the program is a function, called by a special action in the main program.

The parameters passed to rsvparse.pll are a code indicating the command verb and an image of the response line from the retrieval system. The value returned by rsvparse.pll is one of three condition codes, either 0,98,or 99. If the response is parsed without error, the value 0 is returned; if a system or network error message is identified, the value 98 is returned; if the response line cannot be handled by the program, the value 99 is returned.

Rsvparse.pll consists of three major sections corresponding to the three major types of responses. These are responses to (1) set-creating commands, (2) EXPAND table commands, and (3) abstract- and citation-requesting commands. The section executed is chosen according to the verb-code parameter. Responses to the following commands are trivial so they are not parsed: EXPLAIN, BEGIN, DS, LIMITALL, LOGOFF, .FILE, and END. The analysis of the three major sections will now be discussed.

(1) Responses to set-creating commands. The primary function of the response parser in this case is to isolate and store the set's postings count. The parser also compares the returned set number with the set number maintained by IIDA. If the two numbers differ, a warning is issued. If the returned string contains only one numeric value, the line is ignored for parsing purposes; it is assumed to be an intermediate result in response to a super-SELECT command, which is valid under the IIDA-supressed option.

(2) Response lines to the EXPAND command. The elements of all lines but the header are isolated and stored in the student history data base. Both the last lexical and the last thesaural tables are stored in their entirety.

These tables are used by some strategic analyses and by those parse.pll routines which extend SELECT arguments that are references to EXPAND tables. (In the latter case, the references are translated into their corresponding terms.)

(3) Response lines to citation-requesting commands. If the conditions are met for record assistance (see Section 5.2.4) the response lines are saved in a modified form on an external file. If the data base is ERIC (1), COMPENDEX (8), or ONTAP (201), a blank line is recognized as the end of a record. When the response parser recognizes this, it requests the user to rate the relevance of the record to his search objectives. This relevance rating is validated and stored before control returns for the next response line, if any. If the data base is NTIS (6), this recognition technique fails since blank lines are found within records for several formats. With NTIS the format code of the command argument is used in conjunction with the occurrence of certain standard substrings in the record to recognize the blank line that really signals the end of the record. Only at this point is a relevance rating requested.

4.2.3. idverb.pll. The purpose of this program is to do an in-depth verb analysis on command strings that do not begin with any recognized command or command abbreviation. The general form of the program is a function, called by a special action in the main program when an unrecognized command is entered. The value returned is a code indicating the result of the analysis.

Identified by this program are valid DIALOG commands that are not supported by IIDA. These include: .RECALL, .EXECUTE, .RELEASE, and .SORT. Special character abbreviations inherited from RECON by DIALOG are also recognized here. These include: ! for BEGIN, " for EXPAND, # for SELECT, $ for COMBINE,' for TYPE, % for DISPLAY, & for PRINT,) for LIMIT, 0 for PAGE, ? for EXPLAIN, @ for DS, and = for END.

If the command in question is none of the above, the line is scanned for clues that may imply the argument of one or more classes of commands. For instance, if left or right parentheses or a logical operator is present, the command may have been COMBINE; if a question-mark is present, it probably was a truncated SELECT; if a large percentage of the symbols are numbers, it is

likely a TYPE or COMBINE command. In each of these cases, the user is referred to one or more frames of the Exercise 3 file, accessed through HELP, for detailed information on the syntax and use of the assumed commands.

4.3. Strategic Analyses of Commands and Responses

4.3.1. canalysis.pll. This command analysis program is called by the main program. No values are passed to or from canalysis.pll as parameters, although the entire student data base, whose structures are external, is declared in and may be referenced by the program. The program consists of two parts: command-dependent analyses and threshold analyses. The former creates values based upon an analysis of the content and context of the search; the latter compares these analytic values with search history values to determine if strategic conditions exist.

The command-dependent analyses include: isolating and storing a code for the current class of command; counting consecutive commands of the current class; deterimining the average time taken to enter commands; maintaining string and cycle codes and measures; if a new cycle, counting non-referenced sets in the previous cycle; calling normalizer.pll (see Section 4.3.2), if it applies to the current command; computation of command repetition; and computation of argument similarily. Only the last two of these analyses involve algorithms of much complexity. Discussion of these follow.

Computation of command repetition is made only if the command is EXPAND, SELECT, COMBINE, TYPE or DISPLAY. For EXPAND, TYPE, and DISPLAY, the current argument is used for comparison; for SELECT and COMBINE, the normalized form of the current argument is used. The current argument is then compared with all arguments of all previous commands of the same class. Whether the form of these previous arguments is exact or normalized depends again upon the command class. Any matches identified are recorded in the student data base. Both exact (arguments are precisely alike) and virtual (intent is the same but syntax of expression differs) matches are noted.

Computation of argument similarity is performed only upon COMBINE command arguments. Each descriptor or term in COMBINE commands in the current string is compared with each descriptor in the current command's argument. The similarity index of the current command compared with a previous command is computed as: 1/2 x (number of matching descriptors) x (reciprical of descriptor count in the current command + reciprical of descriptor count in the previous command). The sum of these values represents the string's similarity index. This index is high when many of the descriptors are held in common; it is low when few of the descriptors are the same.

Threshold analyses turn on bit-flags to indicate strategic conditions, based on the analytic values generated above and on values stored and accumulated in the student data base. Conditions are identified by threshold analyses in both this program and in ranalysis.pll (see Section 4.3.3). The following conditions are primarily signalled in canalysis.pll: virtual and exact command repetition; excessive string length; increasing count of non-referenced sets; time taken to enter commands; and "thrashing". This last condition is recognized by a low similarity index. A detailed description of all tests made to identify strategic conditions can be found in Appendix 4.

4.3.2 normalizer.pll. The objective of this program is to reduce boolean expressions in command arguments to a standard form, insofar as possible, for comparison purposes. The boolean expressions analyzed are the extended argument of COMBINE and SELECT commands, created by the command parser. (SELECT commands may have boolean extended arguments if the argument refers to a range of entries SELECTed from an EXPAND table.)

Reduction is not always complete on complex boolean expressions. Conversion of the expression to Polish post-fix notation was not done in this instance, although it probably would be a helpful step in future normalization algorithms. On simple expressions, reduction is achieved by applying heuristic algorithms which: (1) change NOT to AND NOT and apply deMorgan's law if necessary; (2) eliminate all parenthesized expressions by distributing AND across ORed terms in parenthetical expressions; (3) alphabetize all factors within terms; and (4) alphabetize all terms within the final complete expression. Each of these algorithms deserve some discussion:

(1) DeMorgan's law is applied whenever the NOT operates on a parenthetical expression. The iterative application of this algorithm eventally replaces all dyadic NOTs with monadic NOTs in the expression. (2) The distributive law is applied to the deepest parenthetical expressions first, eliminating parentheses from the innermost to the outermost set. The argument is scanned both left-to-right and right-to-left, distributing AND across all parenthesized expressions. The resulting expression should be a parenthesis-free string of terms connected by the OR operator. These terms consist of factors which may or may not be connected by the AND operator.

(3) All the factors ANDed together into a term are alphabetized within the term, and placed back into the argument being normalized. (4) Then these terms are alphabetized, connected with ORs, and become the normalized string. This finally normalized string is now available for comparing with other arguments.

For instance, after normalization, the argument of COMBINE (3+4)*(1+2) will match the argument of COMBINE (2+1)*(4+3) in the same search. This virtual repetition may be called to the attention of the user after being signalled by canalysis.pl1, and approved by the Warning Control algorithms of ranalysis.pl1.

4.3.3. ranalysis.pl1. This response analysis program is called by the main program. No values are passed to or from ranalysis.pl1 as parameters, although variables throughout the student data base can be referenced by the program. The program consists of three parts: response-dependent analyses, threshold analyses, and the Warning Control algorithms. The first creates values based upon an analysis of the content and context of the search; the second compares these analytic values with search history values to identify strategic conditions; the third determines whether or not messages about strategic conditions should be sent to the user.

The significant response-dependent analyses include: analysis of null sets (sets with 0 postings); analysis of relevance ratings; and analysis of convergence. All three analyses involve fairly complex algorithms. Discussion of these analyses, the threshold comparisons, and the Warning Control algorithms follow.

Analysis of null sets is done when the command is SELECT, COMBINE, or LIMIT, i.e., a set-creating command. The number of null sets in the search, the cycle, and consecutively, by command, is incremented by 1 if the current set is null. If the null-set-creating command was SELECT with a simple argument, the scopes of all previous EXPAND tables are checked to see if the present argument would have been seen on one or more tables. This is noted by a flag that is part of the current command history.

Analysis of relevance ratings is done when: (1) the command is TYPE or DISPLAY; (2) the format is not 1 (accession numbers only); and (3) IIDA is not suppressed. The relevance rating assigned by the user to the records seen is processed here. The average relevance for records seen of the current set, string, and cycle, are all computed. Also computed is the highest average relevance in the current set, string and cycle.

Analysis of convergence is performed when the command is COMBINE. It results in a code indicating a pattern in the dispersion of set sizes. The size of the current set is compared to the size of both the previous set and the goal set size entered by the user at the beginning of the program. The following conditions are identified with respect to dispersion and stored as a code to be tested later: the current set size is significantly closer to the goal size than the previous set size; the current set size is significantly further away from the goal size than the previous set size; the last two set sizes are static with respect to the goal; the last two set sizes have "bounced" around the goal, or otherwise have a dispersion characteristic that cannot be determined.

Threshold comparison analyses turn on bit-flags to indicate strategic conditions. These follow from a comparison of analytic values generated above and values stored and accumulated in the student history data base. The following conditions are primarily signalled in ranalysis.pll: null set conditions; uninformative format condition; relevance comments and conditions; non-used sets when the user logs off; convergence and dwelling conditions; relevance rating patterns; and null set SELECTion scoped by a previous EXPAND table. A detailed description of all tests made can be found in Appendix 4.

The Warning Control algorithms are all contained in a loop which tests each strategic condition flag in turn. If the flag is off, none of the algorithms are executed. Otherwise, an entry is added to the error history table and the algorithms proceed to determine if threshold adaptations should be made and if a message should be sent to the user. If the strategic condition has a threshold value that is adaptable, the adaptation is made according to the mode. If it is not adaptable and if a message on the condition was seen recently the flag is turned off. Also, if a more specific message is available and signalled, the flag is turned off. Prefix and suffix messages are selected to provide a transition between messages. If the condition's flag is still on, a message about the condition is requested by passing the condition code to mess_out.pl1 (see Section 5.3.3).

5. OTHER SUPPORT PROGRAMS

This section discusses programs that are not immediately concerned with the messages exchanged as part of a search online with DIALOG. Two of these programs, ex1.pl1 and ex3.pl1, operate without placing any outgoing call through the AutoCall device. These are discussed in Section 5.1. Another group of programs represents requests by the user for various types of assistance provided by IIDA. These are discussed in Section 5.2. A third group of programs handles packaged messages. These are discussed in Section 5.3.

5.1. Non-DIALOG Exercises.

Exercises 1 and 3 involve no network link to DIALOG. In some cases, DIALOG responses are simulated in response to user entries that look like DIALOG commands. See Figure 6 for the general flow of these two programs. Both programs access keyed sequential access files. These are derived from strictly sequential files that are, in turn, maintained by the MULTICS text editor. Logical records on the keyed files make up a frame of assistance or instruction. These consist of a variable number of MULTICS records. The key consists of a frame number and a record sequence number within the frame. The instructional intent and content of these files is discussed in Quarterly Report 5.

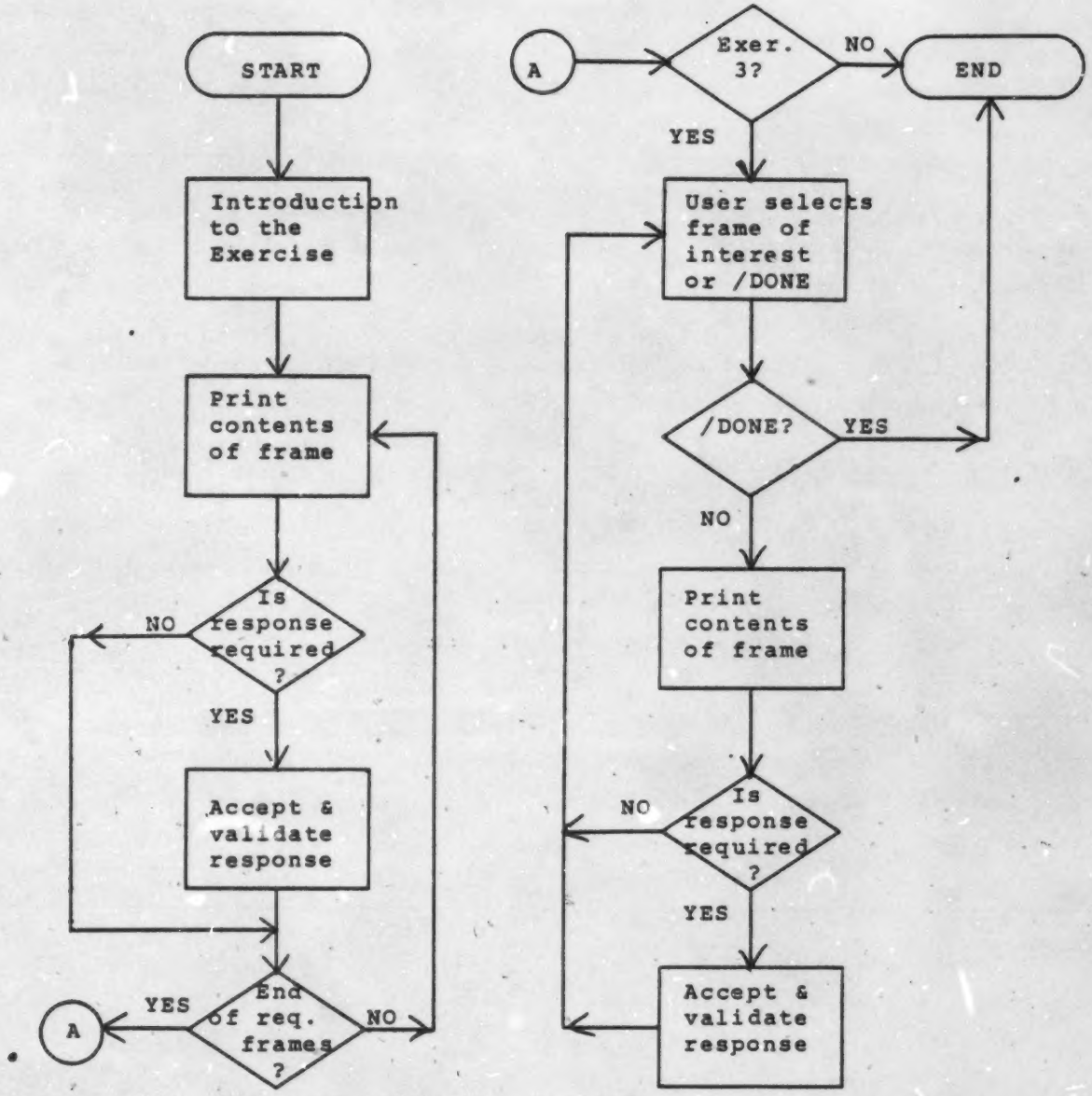

Figure 6. Overview of flow-of-control for Exercises 1 and 3. Note that in Exercise 1 all frames of textual material are required, while in Exercise 3 only the first eight frames are required. After that, the user has his choice of frames to review.

5.1.1. ex1.pl1. After attaching and opening the indexed file, it is essentially printed from beginning to end at the terminal for the user, with some exceptions. If the first non-key character of the record is an asterisk, the record is a request for input from the user. The key of such a record is stored. If the first character is an up-arrow the system pauses for input from the user If the input does not exactly match the rest of the current record or subsequent records that also begin with an up-arrow, the user's entry is invalid. A valid entry is requested again by printing at the terminal the record that began with an asterisk, i.e., the record whose key was just stored.

All characters of user entries are converted from uppercase to lowercase for comparison with acceptable responses, which are all stored in lowercase characters. If the user enters /DONE, he can escape from Exercise 1 and return to the main menu of the IIDA system. If the user's response is not validated after three opportunities for input, the user is also sent back to the main menu. If the first non-key character of the record begins with a tilde, the date and time is printed in the DIALOG format to complete the simulation of a BEGIN or LOGOFF command.

5.1.2 ex3.pl1. This program operates in two modes: (1) as a stand-alone exercise; and (2) as the source of IIDA's online HELP that discusses commands and techniques. Only the first use of this program will be discussed here; the use of ex3.pl1 as HELP is discussed in Section 5.2.2. In many ways, the operation of this program as an exercise looks like the operation of ex1.pl1. The data file for Exercise 3, however, involves a highly structured system of keys. The data structure of the Exercise 3 file thus deserves an explanation which will precede the explanation of the processing of this file.

Frames again consist of a group of MULTICS records whose keys are the frame number followed by a line sequence number within the frame. The frames are numbered so that the instructional material logically forms an n-ary tree. Frame 0 is the root of this tree. Single digit frames form the first level; double-digit frames form the second level; etc. With as many as 5 digits to a frame number, the tree may descend to 5 levels. More detail on any subject can be found in frames whose initial digits are the current frame

number, and whose last digit identifies a detail frame, if such a frame exists. The text of each frame contains references to more detailed frames on the same subject, if such detailed frames exist.

Exercise 3 requires the users to read frames 0 through 7. These essentially serve as a table of contents to the rest of the information in the file. When these first 8 frames have been displayed, the user is allowed to explore the remainder of the file by entering frame numbers. It is expected that he will follow one or more branches for more information on the search process. The user can return at any time to the system's main menu by entering /DONE.

As in the Exercise 1 file, the first character of the machine record following the key may be a control character. If it is, it is not printed at the terminal, but is acted upon by the program. Remaining characters in the record may be printed. If the control character is the percent sign, the record contains simply the frame number. The program waits for an entry or a carriage return at the end of each frame. If the first character is an asterisk, the line requests a response from the user. One or more subsequent line begin with an up-arrow and contain the exact text of allowed responses. Processing continues when a correct answer is given. If the line begins with "$Entering", it is not printed for Exercise 3, but is printed when the file is used as HELP since subsequent lines simulate a DIALOG response to a user's simulated DIALOG command. If it begins with a tilde, the date and time is interpretted in the DIALOG format.

5.2. User-Initiated Assistance

5.2.1 help-iida.pll. This program provides two overt types of assistance: (1) access to the frames of the Exercise 3 file while online to DIALOG; and (2) up-to-the-minute reviews of the search from various perspectives. This programs also serves as the vehicle through which IIDA invokes Exercises 1 and 3 as exercises, although this latter use of help iida.pll is transparent to the user.

HELP is invoked by the user entering either /HELP or /Hn, where n is a number between 1 and 7. Entry of /HELP alone returns a menu summarizing seven types of help available to the user. Entry of /Hn goes directly to HELP type n. Once in HELP, the user may see the other types of HELP by entering the numbers, or return to the search by entering a DIALOG command. Entering /H1 accesses frame 0 of the Exercise 3 file; entering /H1.m accesses frame m of the file. For use of this file as HELP, see Section 5.2.2.

Histories of the search also can be requested by number. The HELP options are: (1) instructional frames, (2) command history, (3) set history, (4) records viewed history, (5) error history, (6) descriptors history, and (7) quick advice. Tables in response to helps (2) through (6) appear exactly as they do when proc_rep.pl1 is invoked (see Section 3.4.1). Note, however, that a user cannot request histories of strategic conditions or help usage, as the proctor can.

Option 7 provides quick advice. If a strategic error was raised by the previous DIALOG command, quick advice can be requested by entering either /H7 or /QA. A sentence or two recommending a course of action is printed for each strategic condition raised by the previous search command. These advice sentences are store in messea.pl1 (see Section 5.3.1).

5.2.2 ex3.pl1 as help. When instructional frames are requested through HELP during a search, the Exercise 3 file is attached and opened. If the HELP request included a frame number, e.g., /h1.231, the first line of that frame is accessed and processing begins. If the request did not include a frame number, e.g. /H1, the first line of frame 0 is accessed and processing begins.

Processing includes printing each line of the frame. If the line begins with a tilde, the date and time is printed in the format given by DIALOG in response to a BEGIN or LOGOFF command. If a line requires or verifies a user's response, it is ignored, since that pertains to the file being used strictly as an exercise. After the frame has been printed, the user is asked to enter a frame number or a search command. If a search command or any IIDA-directed command is entered, control returns to the main program with the command for processing. If another frame number is entered, the frame is processed as described above.

If the user enters /MENU n, where n is a frame number, only the menu part of the frame is printed. These are lines that begin with a dollar-sign as a control character and which refer by number to frames that contain more detailed information on the subject discussed in the current frame.

5.2.3. da.pl1. This program was written to provide a user with assistance in choosing descriptors. Both it and its successor, rec-assist.pl1, were experimental efforts, and were not part of the software evaluation experiments that will be reported in subsequent Quarterly Reports. Da.pl1 is called by the main program, iida.pl1, through a special action invoked when the user enter the IIDA-directed command, /DA. Assistance is provided by IIDA automatically sending to DIALOG: (1) a full format TYPE command, relayed to user in full or in part, or (2) an EXPAND command whose argument is a keyword or phrase supplied by the users. Figure 7 presents a flowchart of this program, which is discussed in more detail below.

After invoking this subsystem, the user is given a menu from which he may choose one of three usage modes. Mode 1 requires the user to choose a set he thinks best represents his search objectives. IIDA determines if records from this set have been seen yet in the full format. If they have been, descriptors are isolated and printed from records in the set not yet seen by the user. This may suggest descriptors helpful for the search. If records of the set have not yet been seen in the full format, up to four records from the set are printed in that format. Descriptors in the record text may be found useful for the search.

Mode 2 provides the EXPAND table type of assistance. The user is asked to enter keywords of phrases related to the objectives of the search. IIDA determines if any keyword has already appeared within the scope of a previous EXPAND table, that is, if the keyword falls alphabetically between the first and last entry of any previous table. If so, the user is refered to that table. If not, IIDA offers to issue an EXPAND on the terms, one at a time.

Mode 3 operates much like Mode 1, except that IIDA, rather than the user, chooses the set number. IIDA identifies the last set created, if any, with an average relevance greater than 3. If there is one, that set becomes the set

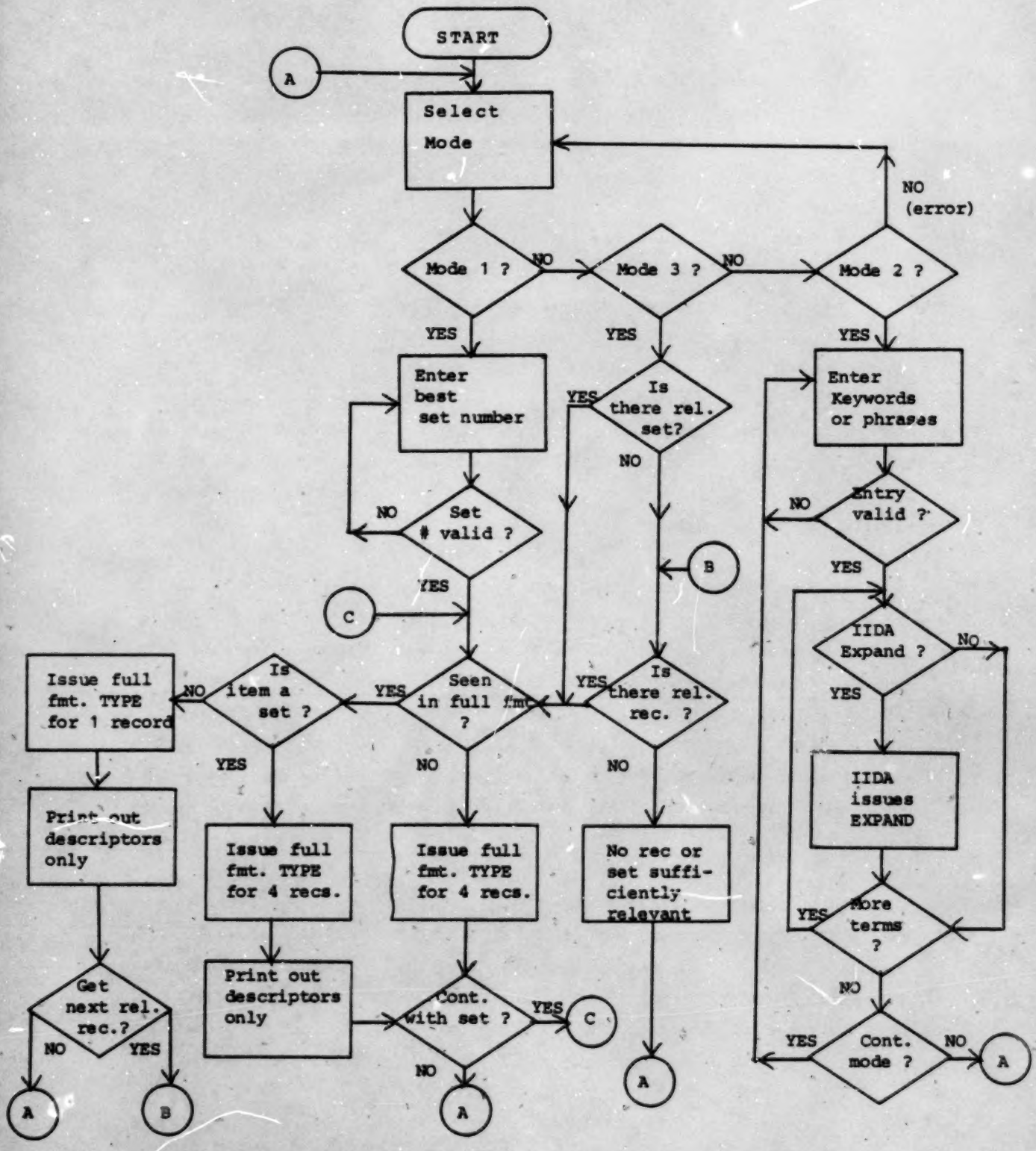

Figure 7. Flowchart of Descriptor Assistance. Note that the user may return to the main program at any time when this program seeks input from the user, by entering any DIALOG- or IIDA-directed command.

for descriptor assistance activity. If there isn't one, IIDA searches for the last record viewed, if any, with an average relevance greater than 3 or any previous set with a higher average relevance than such a set. If there is such a record, that record's set becomes the set for descriptor assistance activity, much as is done with Mode 1. Mode 1 differs from Mode 3 in that the user supplies the set number in Mode 1 while IIDA identifies the set number in Mode 3.

After each IIDA response, the user is asked if he wishes to continue in the current mode. If he does, the mode will be continued if there is additional input to drive the mode. If he doesn't or if there is no driving input, he is returned to the top menu from which descriptor assistance modes may be selected. Thus, the user may continue in Mode 1 as long as he furnishes valid set numbers, in Mode 2 as long as keywords or phrases are provided by the user, and in Mode 3 as long as sets or records with the required relevance are found.

The user may leave descriptor assistance at any time by entering any DIALOG- or IIDA-directed command. This allows the user to resume his search as soon as he find descriptors to his liking. This exit technique is also consistent with the technique found in other forms of IIDA assistance, such as HELP and record-assistance.

5.2.4 rec-assist.pl1. This program was also written to assist users in finding records relevant to their search topics. Rec_assist.pl1 is an external procedure called as a function by a main program special action, and is not passed any parameters. However, it does return a character string which may be either the null string (the user took rec_assist.pl1 to completion) or a command directed to DIALOG (the user terminated rec-assist.pl1 short of completion). Structurally, rec_assist.pl1 is quite complex. It contains 10 internal procedures which perform activities such as building and searching a balanced binary tree, and performing recursive quicksorts and binary searches of various arrays used internally.

Before this experimental program is invoked, the response parser (rsvparse.pl1), under certain conditions, creates an external file containing a modified copy of records returned by DIALOG in response to a TYPE command. This is done and the file is kept when the following conditions obtain: (1) the user-id is one restricted for use by the IIDA-staff; (2) the data base is COMPENDEX (file 8); (3) records were requested in full format (format 5); (4) the set contains at least 12 records; and (5) at least one of the records viewed was assigned a relevance greater than 3. These records are analyzed by rec_assist.pl1. They will be referred to as the "sample" in the discussion that follows.

When the user has finished seeing and rating records under the above conditions, rec_assist.pl1 is automatically invoked. The file of modified bibliographic records is closed for output and opened for input. IIDA then begins a weighted word-frequency analysis on the sample. This includes the following:

Every significant word in the title, abstract, descriptor, and identifier fields of the records is isolated. A stop-list of nearly 200 words is checked to eliminate trivial words. The words are maintained at nodes of a balanced binary tree. The nodes of this tree are keyed by the initial 7-characters (trunks) of significant words. If more than one word has the same trunk, the full words are linked together in a list anchored to the trunk's node. The tree is searched for every significant word isolated. If the word's trunk is not found, a node is inserted, and the tree is rebalanced if necessary. When found or established, the number of occurences of the trunk is registered at the node by record number.

After the file is completely processed, the weighted relevance of each trunk is computed as follows: All nodes of the tree are read. For each node a weight is computed as the sum of the products of record relevance ratings by the number of occurences in the those records. The addresses of the 15 nodes with the highest computed weights are saved in an array. The weight range of these high weight nodes is mapped into a range of 1 to 5.

The user is given at this point the option of returning to the search by entering a DIALOG command. If he decides instead to continue with the

experiment, the 15 selected words, both in their truncated and all entire forms, are printed in alphabetical order at the terminal. The user then has the opportunity to add to the list new terms with user-assigned weights, to delete from the list unsuitable terms, and to change the assigned weight of terms already on the list. The user may choose to see all the terms again, in weight order, and to recycle through the list revision process.

When the user is satisfied with the list as it stands, he is given three options: (1) he may terminate the experiment and return to the search by entering a DIALOG command; (2) he may have IIDA select records through an algorithm that emphasizes precision; or (3) he may have IIDA select records through an algorithm that emphasizes recall.

The algorithm of the precision experiment will do the following: (1) issue a TYPE command in format 5 on the next 12 records in the set, or on the rest of the records in the set, if there are less than 12 records; (2) isolate significant words, much as was done with the initial sample file; (3) determine for each significant word if it has a 7-character trunk that matches a trunk on the modified weighted word list; (4) if there is a match, add the weight score into an accumulating score for the record; (5) when through scanning the 12 records, print in format 5 the four records with the highest cumulative scores; (6) request a relevance rating for each of these 4 records; (7) average and report the relevance of the initial sample and the records subsequently selected by IIDA; and (8) return to the search.

The algorithm of the recall experiment will do the following: (1) issue the TYPE command to DIALOG for up to 100 record accession numbers from the sample set (these are saved in an array); (2) SELECT as a truncated argument each trunk from the revised list; (3) COMBINE, in turn, each of the resulting sets with the initial sample set using AND-logic; (4) isssue TYPE commands to DIALOG to retrieve the accession numbers for each set resulting from the combination; (5) look up each returned accession number on the table of accession numbers saved from the TYPE response (1); (6) if the accession number is found, add to an accumulator for the accession number the weight of the associated revised list term; (7) when the analysis is through, print in format 5 the four records whose cumulative scores are the highest; (8) request a relevance rating for each of these 4 records; (9) average and report the

relevance of the initial sample and the IIDA-selected records; and (10) return to the main program.

5.3. Message Handlers

At least three types of packaged messages may be sent to the user. These include messages that are general in nature (Section 5.3.1), messages that are specific to syntactic errors (Section 5.3.2), and messages that are specific to strategic conditions (Section 5.3.3).

5.3.1. messen.pll. This program provides IIDA with a facility for communicating strictly canned messages of a general nature to the user, based upon a message code passed by the calling procedure. Messen.pll is normally called by the main program, with the message code given in a production rule. Other programs, however, call messen.pll, on occassion. The messages are set up as data in a table, with controlling information set up in an auxillary table. Execution of the program involves accessing and printing a message from the table, as requested by the message code and directed by the auxillary table.

Messages are organized in the table according to their function in the system. Message 1-6 provide introductory frames to the IIDA system. Message 7 gives the initial menu of exercises and modes. Message 8 introduces the Exercise 2 search topic. Message 9 introduces the help library (Exercise-3-as-help). Messages 26-30 are menus and other canned frames of /HELP. Messages 49-50 explain why PRINT and a second BEGIN in Exercise 2 are not allowed. Messages 51-86 amplify error messages received from DIALOG and the network. Messages 91-98 provide prefixes to strategic condition messages. Messages 150-200 provide advice on strategic conditions, if requested through /HELP or /QA. Messages 201-208 provide suffixes for strategic condition messages. Messages codes not specified above are either obsolete or don't exist.

5.3.2 mesyner.pll. This program describes syntactic error conditions and "hybird" conditions. The latter are conditions caught by the command parser but are properly classified as strategic. Approximately 100 error messages are reconstructed by this program. The base of these messages is

given in a data table at the beginning of the program. The student history data base is updated by mesyner.pll with information on syntax errors and hybrid conditions. See Appendix 3 for the base syntax error messages. Also, see Figure 8 for a classification scheme for these messages.

The hybrid conditions are handled first. These include: (1) reference to a null set in a COMBINE argument and excessive truncation (the commands are sent to DIALOG); and (2) reference to a null set in a TYPE or LIMIT argument and too many sets requested for a given format (the commands are not sent to DIALOG). No strictly syntactic errors are transmitted to DIALOG. Rather, they are reconstructed for the user as part of an error message. The assumed command and argument are indicated, and the point up to which parsing was successful, if any, is printed. The error message from the table follows this analysis.

For instance, a typical syntactic command error message is:

**SYNTAX ERROR:: Command assumed: 'combine'; argument assumed:
'1 and oil'. The argument is acceptable up through '1 and '.
However, the next character is illegal in a COMBINE command.

5.3.3. mess-out.pll. This program constructs messages for the user which describe strategic conditions signalled by the two analysis programs and approved for display to the user by the Warning Control algorithms. The program is called by ranalysis.pll, at the very end of the Warning Control loop (see Section 4.3.3). It receives as a parameter from ranalysis.pll the code of the strategic condition which is to be called to the attention of the user.

This program consists entirely of a set of comparisons of the current code against constants representing the 45 possible codes. If the comparison indicates a match, a message for the condition is constructed and sent to the user. Control then returns to the Warning Control loop in ranalysis.pll. The message is constructed out of constant character strings and values in the student history data base that pertain to the current strategic condition. See Appendix 4 for a list of strategic conditions in code order.

Code	Commands / Comments
Syntactic Error Codes	
8	SELECT: argument consists entirely of truncation symbols
9	END: SDI option requested, prohibited by the IIDA system
10-11	.FILE & BEGIN: argument is data base prohibited by IIDA
12	END: some argument is given, which is invalid
13	LOGOFF: some argument is given, which is invalid
14	DISPLAY SETS: some argument is given, which is invalid
16-17	PAGE command errors
20-29	EXPAND command errors
30-50	SELECT command errors
51-69	COMBINE command errors
70-79	LIMIT command errors
80-96	TYPE & DISPLAY command errors
97	all commands: missing argument when required
98	EXPLAIN command error: argument not explained by DIALOG
99	IIDA programming error
Hybrid Error Codes	
1	too many records requested for given format - an IIDA restriction
101, 104	COMBINE argument refers to a null set
102	TYPE argument refers to a null set
103	LIMIT argument refers to a null set
105	excessive truncation: prior to argument's 4th character
110	BACKSPACE used in argument - a valid character cancel symbol for DIALOG, but not valid in the MULTICS environment

Figure 8. Classification of syntactic and hybrid error codes. Base messages for every error code is given in Appendix 3. Note that in the proctor report, where the errors are reported, the constant 400 is added to the syntactic codes in order to distinguish them from the strategic conditions also reported there.

FOOTNOTES

(1) Meadow, C. T., et al., Individualized Instruction for Data Access (IIDA): Quarterly Reports Nos. 1 through 5. NSF Grant No. DSI-77-26524. Philadelphia: Drexel University and the Franklin Research Center, June 1978 through June 1979.

(2) Marcus, R. S., and Reintjes, J. F., Computer Interface for User Access to Heterogeneous Information Retrieval Systems. NSF Grant No. SIS-75-22946. Cambridge, Mass.: Massachusetts Institute of Technology, April 1977.

(3) Meadow, et al., Quarterly Report No. 1, pp. 6-9.

(4) Marcus and Reintjes, pp. 10-26.

(5) Meadow, et al., Quarterly Report No. 2, p. 20.

(6) Marcus and Reintjes, pp. 10-26.

APPENDIX 1 — SAMPLE PROCTOR REPORT

40861210j 08/21/79 2035.7

*** COMMAND HISTORY ***

CMD	CLASS*	TYPE**	SET	STR	GRP	STK	COMMAND	ARGUMENT
1	O	10	0	1	1	0	begin	1
2	O	23	1	2	1	0	select	computer programming
3	E	40	0	0	0	0	type	1/5/1
4	E	40	0	0	0	0	type	1/5
5	E	40	0	0	0	0	type	1/50
6	E	40	0	0	0	0	type	1/5/0
7	O	23	2	2	1	0	select	computer
8	E	40	0	0	0	0	type	2/5/1-5
9	O	40	0	3	1	0	type	2/5/1
10	E	48	0	0	0	0	page	mathematical models
11	O	48	0	3	1	0	page	
12	O	20	0	1	2	0	expand	mathematical models
13	O	24	3	1	2	0	select	e6
14	O	40	0	2	2	0	type	3/4/1-3
15	O	48	0	2	2	0	page	
16	E	24	0	0	0	0	select	e21
17	O	23	4	1	3	0	select	mathematical statistics
18	O	23	5	1	3	0	select	mathematical structure
19	O	40	0	2	3	0	type	5/4/1-2
20	O	23	6	1	4	0	select	mathematical theory
21	O	40	0	2	4	0	type	6/5/1-2
22	O	23	7	1	5	0	select	problem solving
23	E	40	0	0	0	0	type	7/4/1-10
24	E	40	0	0	0	0	type	7/5/1-10
25	O	40	0	2	5	0	type	7/4/1-6
26	O	23	8	1	6	0	select	mathematics
27	E	40	0	0	0	0	type	8/8/1-7
28	O	40	0	2	6	0	type	8/8/1-3
29	O	23	9	1	7	0	select	computing
30	O	20	0	1	7	0	expand	computing
31	O	23	10	1	7	0	select	computest
32	O	40	0	2	7	0	type	10/4/1
33	O	40	0	2	7	0	type	9/4/1-3
34	O	23	11	1	8	0	select	computes
35	O	40	0	2	8	0	type	11/4/1-2
36	O	15	0	0	0	0	logoff	

*** SET HISTORY ***

CMD	SET NUM	SET SIZE	REFS	AVG. REL.	COMMAND	EXTENDED ARGUMENT
2	1	0	2	0.00	select	computer programming
7	2	11726	2	5.00	select	computer
13	3	1675	1	5.00	select	mathematical models
17	4	0	0	0.00	select	mathematical statistics
18	5	2	1	3.00	select	mathematical structure
20	6	2	1	3.00	select	mathematical theory
22	7	5286	3	4.00	select	problem solving
26	8	14162	2	4.66	select	mathematics
29	9	647	1	4.66	select	computing
31	10	1	1	5.00	select	computest
34	11	30	1	4.00	select	computes

*** SETS & RECORDS VIEWED ***

CMD	SET	FMT	FIRST REC	LAST REC	AVG REL	ARGUMENT
9	2	5	1	1	5.00	2/5/1
14	3	4	1	3	5.00	3/4/1-3
19	5	4	1	2	3.00	5/4/1-2
21	6	5	1	2	3.00	6/5/1-2
25	7	4	1	6	4.00	7/4/1-6
28	8	8	1	3	4.66	8/8/1-3
32	10	4	1	1	5.00	10/4/1
33	9	4	1	3	4.66	9/4/1-3
35	11	4	1	2	4.00	11/4/1-2

*** ERROR HISTORY ***

CMD	CODE	TYPE	COMMAND	ARGUMENT
3	87	SYNTAX	type	1/5/1
4	102	SYNTAX	type	1/5
5	84	SYNTAX	type	1/50
6	102	SYNTAX	type	1/5/0
8	435	STRATEGY	type	2/5/1-5
10	16	SYNTAX	page	mathematical models
16	45	SYNTAX	select	e21
19	426	STRATEGY	type	5/4/1-2
21	426	STRATEGY	type	6/5/1-2
21	427	STRATEGY	type	6/5/1-2
23	435	STRATEGY	type	7/4/1-10
24	435	STRATEGY	type	7/5/1-10
27	435	STRATEGY	type	8/8/1-7

*** STRATEGIC ERRORS BY CODE ***

ERROR CLASS	ERROR CODE	TOTAL TIMES	LAST CMD	CLASS DESCRIPTION
8	26	2	21	
8	27	1	21	
8		3		Relevance comments
10	35	4	27	
10		4		Viewing request excessive

*** DESCRIPTORS HISTORY ***

TIMES USED SEL	EXP	REL	COM	DESCRIPTOR
1	0	0	0	computer
1	0	0	0	computer programming
1	0	0	0	computes
1	0	0	0	computest
1	1	0	0	computing
1	1	0	0	mathematical models
1	0	0	0	mathematical statistics
1	0	0	0	mathematical structure
1	0	0	0	mathematical theory
1	0	0	0	mathematics
1	0	0	0	problem solving

*** HELP HISTORY ***

TIME IN	TIME OUT	TIME DIFF	HELP OPTION 01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
2001.7	2003.2	1:32	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
2003.2	2004.7	1:29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2005.6	2006.3	0:39	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2021.2	2021.8	0:36	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
2021.8	2022.5	0:41	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

APPENDIX 2 -- SAMPLE ADAPTIVE VARIABLE REPORT

THRESHOLD ADAPTATIONS

CLASS	NET CHANGE	FINAL VALUE	INITIAL VALUE	ADAPTATIONS	ADAPTED VARIABLE
essential command repetition	0	0	0	0	1
	3	5	2	1	2
string length	0	7	7	0	3
	12	17	5	5	4
	0	10	10	0	5
null sets	0	2	2	0	6
	0	3	3	0	7
	0	5	5	0	8
non-used sets	6	9	3	2	9
	0	5	5	0	10
exact command repetition	0	0	0	0	11
	0	2	2	0	12
	0	6	6	0	13
dual command repetition	0	0	0	0	14
	0	2	2	0	15

```
/* m. 8 */ 'an argument which is entirely truncation cannot be processed.',
/* m. 9 */ 'SDI operations are not supported by IIDA software.',
/* m.10 */ 'only data base numbers 1, 6, 8, and 201 are allowed by IIDA.',
/* m.11 */ 'it can only be a data base number, either  1, 6, 8, or 201.',
/* m.12 */ 'END does not take an argument.',
/* m.13 */ 'LOGOFF does not take an argument.',
/* m.14 */ 'DISPLAY SETS takes no arguments.',
/* m.15 */ '',
/* m.16 */ 'PAGE takes no argument.',
/* m.17 */ 'the PAGE command must immediately follow a valid EXPAND command.',
/* m.18 */ '',
/* m.19 */ '',
/* m.20 */ 'the equals sign does not follow a valid 2-character prefix code.',
/* m.21 */ 'the prefix is not a valid 2-character code in this data base.',
/* m.22 */ 'there was no previous EXPAND, which is necessary when referencing an E-number.',
/* m.23 */ 'the next E-number exceeds the highest E-number on the last EXPAND table.',
/* m.24 */ 'for an argument in the E-number form, the character that follows is not legal.',
/* m.25 */ 'there was no prior EXPAND on related terms so an R-number can't be referenced.',
/* m.26 */ 'the next R-number exceeds the highest R-number on the EXPAND table.',
/* m.27 */ 'for an argument in the R-number form, the character that follows is not legal.',
/* m.28 */ 'parentheses for a thesaurus lookup are not matched or are misplaced.',
/* m.29 */ 'there are no related terms in the thesaurus for this EXPAND entry.',
/* m.30 */ 'the next character is not legal in the context of a list of E- and R-numbers.',
/* m.31 */ 'the next character does not represent a valid infix format.',
/* m.32 */ 'the attempted truncation is not legal with the infix or suffix format.',
/* m.33 */ 'the infix form of the SELECT argument is incomplete or incorrect here.',
/* m.34 */ 'SELECTion from an R-table is assumed, but there is no prior thesaurus EXPAND.',
/* m.35 */ 'the next R-number is higher than the highest R-number on the table.',
/* m.36 */ 'the next character does not complete an R-number range SELECTion.',
/* m.37 */ 'when SELECTing from an R-table, the next character should be a number.',
/* m.38 */ 'the next number is less than the first number SELECTed from an R-table.',
/* m.39 */ 'both a prefix and a suffix are implied, which should not occur together.',
/* m.40 */ 'the equals sign does not follow a valid 2-character prefix code.',
```

130

129

```
/* m.41 */ "the prefix is not a valid 2-character code for this data base.",
/* m.42 */ "the slash implies use of a suffix but is not followed by a valid suffix code.",
/* m.43 */ "use of a suffix is implied but the one given is not valid for this data base.",
/* m.44 */ "SELECTion from an E-table is assumed, but there is no previous EXPAND.",
/* m.45 */ "the next E-number is larger than the highest E-number on the EXPAND table.",
/* m.46 */ "the next character does not complete an E-number range SELECTion.",
/* m.47 */ "if SELECTing from an E-table, the next character should be a number.",
/* m.48 */ "the next number is less than the first number SELECTed from an E-table.",
/* m.49 */ "the next character is invalid for a SELECTion from an EXPAND table.",
/* m.50 */ "a SUPER SELECT is implied, which is not managed by IIDA software.",
/* m.51 */ "the next left parenthesis is illegally placed.",
/* m.52 */ "the next right parenthesis is illegally placed.",
/* m.53 */ "it contains more right parentheses than left parentheses.",
/* m.54 */ "the next AND operator is illegally placed.",
/* m.55 */ "the next OR operator is illegally placed.",
/* m.56 */ "the next NOT operator is illegally placed.",
/* m.57 */ "the next character or operator is illegal in a COMBINE command.",
/* m.58 */ "the set number that follows is illegally placed.",
/* m.59 */ "the next set number is zero or exceeds the largest set number in this search.",
/* m.60 */ "it does not begin with a number, as required.",
/* m.61 */ "the next character is illegal in a COMBINE command of the form: m-n/op.",
/* m.62 */ "the next character is illegal in a COMBINE command of the form: m-n/op.",
/* m.63 */ "the next character is illegal in a COMBINE command of the form: m-n/op.",
/* m.64 */ "the next set number exceeds the largest set number in the search.",
/* m.65 */ "the second set number is smaller than the first set number.",
/* m.66 */ "the operator is neither AND nor OR, which only are legal in this context.",
/* m.67 */ "it contains more left parentheses than right parentheses.",
/* m.68 */ "it is incomplete.",
/* m.69 */ "it does not contain the required minimum of 2 sets.",
/* m.70 */ "it does not begin with a set number.",
```

131

132

```
/* m.71 */ 'the set number is zero or exceeds the largest set number in the search.',
/* m.72 */ 'the next character should be '/', to delimit the set number in the argument.',
/* m.73 */ 'the next number is assumed to be an accession number, which must be 6 digits.',
/* m.74 */ 'the accession numbers must be separated by a dash.',
/* m.75 */ 'a second accession number is expected to follow at this point.',
/* m.76 */ 'the next number is assumed to be an accession number, which must be 6 digits.',
/* m.77 */ 'the second accession number must be greater than the first accession number.',
/* m.78 */ 'the second accession number must be followed by the slash (/) delimiter.',
/* m.79 */ 'use of a suffix is implied, but the one used is illegal for this data base.',
/* m.80 */ 'it does not begin with a set number.',
/* m.81 */ 'the set number is 0 or is greater than the maximum set number in this search.',
/* m.82 */ 'the character following the set number should be a slash (/).',
/* m.83 */ 'the next character should be a format code, which must be a number.',
/* m.84 */ 'the next character should be a format code which must be between 1 and 8.',
/* m.85 */ 'the character following the format code should be a slash (/).',
/* m.86 */ 'a record number is expected to follow, but does not.',
/* m.87 */ 'the next record number is greater than the total number of records in the set.',
/* m.88 */ 'the next characters are illegal in the context of a record-number range.',
/* m.89 */ 'a record number is expected to follow, but does not.',
/* m.90 */ 'the next number is smaller than the first number.',
/* m.91 */ 'the next number exceeds the number of records in the set.',
/* m.92 */ 'the remaining characters are superfluous.',
/* m.93 */ 'the next number, assumed to be an accession number, is too large.',
/* m.94 */ 'the next number, assumed to be an accession number, is not followed by a '/'.',
/* m.95 */ 'a format number between 1 and 8 is expected next.',
/* m.96 */ 'if no record numbers are designated, only the first record will be TYPEd.',
/* m.97 */ 'this command requires an argument, which is missing altogether.',
/* m.98 */ 'the concept cannot be explained by the DIALOG EXPLAIN command.',
/* m.99 */ 'IIDA CONTAINS A PROGRAMMING ERROR; PLEASE NOTIFY THE IIDA PROCTOR.',
```

APPENDIX 4 -- STRATEGIC CONDITIONS - BY CODE

DELETED

Rule 1:

if the number of essential repetitions thus far in the search (rep_knt_search(2)) is less than or equal to 2 and at least one such repetition occurs during the current cycle (rep_knt_cycle(2)) then a message is signalled (te_send(1)) which gives the repeating commands.

Rule 2:

if the number of essential repetitions thus far in the search (rep_knt_search(2)) is 1 or less and at least one such repetition occurs during the current cycle(rep_knt_cycle(2)) then a message is signalled (te_send(2)) which gives the repeating commands and help is offered.

Rule 3:

if the EXPAND command occurs 8 times consecutively(cons_now_type) then a message is signalled (te_send(3)).

Rule 4:

if the SELECT command occurs 8 times consecutively(cons_now_type) then a message is signalled (te_send(4)).

Rule 5:

if type 2 commands(EXPAND/SELECT) occur 8 times consecutively (cons_now_type) then a message is signalled (te_send(5)).

Rule 6:

if the COMBINE command occurs 5 times consecutively (string_length(s_last,st_last)) then a message is signalled (te_send(6)).

Rule 7:

if the TYPE command occurs 5 times consecutively (string_length(s_last,st_last)) then a message is signalled (te_send(7)).

Rule 8:

if any string of commands is 10 (string_length(s_last,st_last)) and the major command type (c_type_maj(c_last)) is > 4 then a message is signalled (te_send(8)) .

Rule 9:
if two consecutive nulls occur from the use of the SELECT command(zero_knt_cons(1)) then a message is signalled (te_send(9)).
RULE FUNCTIONS ONLY ONCE IN EACH CYCLE

Rule 10:
if two consecutive nulls occur from the use of the COMBINE command(zero_knt_cons(2)) then a message is signalled (te_send(10)).
RULE FUNCTIONS ONLY ONCE IN EACH CYCLE

Rule 11:
if two consecutive nulls occur(zero_knt_cons(3)) then a message is signalled (te_send(11)).
RULE FUNCTIONS ONLY ONCE IN EACH CYCLE

Rule 12:
if three nulls occur in a cycle from the use of the SELECT command(zero_knt_cycle(1,s_last)) then a message is signalled (te_send(12)), the arguments given for the nulls, those arguments which have occurred within the scope of a previous EXPAND table are listed separately.

Rule 13:
if three nulls occur in a cycle from the use of the COMBINE command(zero_knt_cycle(2,s_last)) then a message is signalled (te_send(13)) which gives the arguments of the nulls.

Rule 14:
if three nulls occur in a cycle(zero_knt_cycle(3,s_last)) then a message is signalled (te_send(14)) which gives the arguments of the nulls.

DELETED
Rule 15:
if the total nulls thus far in the search(zero_knt_search(1)) from the use of the SELECT command is 5 then a message is signalled (te_send(15)), the arguments given for the nulls, those arguments occurring within the scope of a previous EXPAND table are listed, and the IIDA expand is issued.

DELETED
Rule 16:
if the total nulls thus far in the search(zero_knt_search(2)) from the use of the COMBINE command is 5 then a message is signalled (te_send(16)) which gives the arguments of the nulls.

DELETED
Rule 17:
if the total nulls thus far in the search(zero_knt_search(3)) is 5 then a message is signalled (te_send(17)) which gives the arguments of the nulls.

Rule 18:
if the record format of the current TYPE command is 1 (record_format(r_last) then a message is signalled (te_send(18)).

Rule 19:
if the number of non-used sets created during the last cycle is 3 (no_ref_cycle) then a message is signalled (te_send(19)) which gives the arguments of the non-used sets.

DELETED
Rule 20:
if, upon entering the last cycle, the number of non-used sets created thus far is 5 (no_ref_total) and this total is not reduced by the end of the last cycle(no_ref_total - no_ref_cycle <=0) then a message is signalled (te_send(20)) which gives the arguments of the non-used sets.

DELETED
Rule 21:
if the number of exact repetitions thus far(rep_knt_search(1)) is 2 or less and at least one such repetition occurs during this cycle then a message is signalled (te_send(21)).

Rule 22:
if the number of exact repetitions thus far(rep_knt_search(1)) is 1 or more and at least one such repetition occurs during this cycle then a message is signalled (te_send(22)).

Rule 23:
if the total command repetitions thus far(rep_total) is 6 or more then a message is signalled (te_send(23)) .

Rule 24:
if for at least 4 consecutive COMBINE commands using the AND/OR operators, the similarity index(sim_avg <= t1) is less than .25 then a message is signalled (te_send(24)) which gives the arguments of the COMBINE commands .

Rule 25:
if for at least 6 consecutive COMBINE commands using the AND/OR operators, the similarity index(sim_avg >t2) is greater than .75 and the set size is conversing toward the goal (set_size_disp = 1) then a message is signalled (te_send(25)).

Rule 26:
if the average relevance of documents viewed at this command (view_avg(r_last)) is less than 3 (t3) and more than 2 records are seen then a message is signalled (te_send(26)).

Rule 27:
if for at least 4 consecutive TYPE commands the average relevance (fudge_rel - 25 > view_avg(r_last)) is decreasing then a message is signalled (te_send(27)).

Rule 28:
if maximum relevance of a group in the previous cycle(max_rel(g_last-1)) is greater than the maximum relevance of the last cycle(max_rel(g_last)) then a message is signalled (te_send(28)).

Rule 29:
if the average relevance at this command(view_avg(r_last)) is greater than 4 (t4) and more than 2 records are seen then a message is signalled (te_send (29)).

Rule 30:
if the current command is LOGOFF(cmd_ch_code(c_last)) and a set exists which was not viewed with a full format but has a relevance rating of 4 or more then a message is signalled (te_send(30)) the relevant set given and the user is allowed to resume his search.

Rule 31:
if for at least 4 consecutive COMBINE commands using the AND/OR operators the similarity index is greater than .75(sim_avg>=t2) and the set size is diverging from the goal(set_size_disp = 2) then a message is signalled (te_send(31)).

Rule 32:
if for at least 5 consecutive COMBINE commands using the AND/OR operators the similarity index is greater than .75(sim_avg>=t2) and the set size is static relative to the goal(set_size_disp = 3) then a message is signalled (te_send(32)).

Rule 33:
if for at least 5 consecutive COMBINE commands using the AND/OR operators the similarity index is greater than .75(sim_avg>=t2) and the set size relationship to the goal cannot be determined (set_size_disp = 4) then a message is signalled (te_send(33)).

Rule 34:
if the average time taken to enter commands(time_avg) is greater than 30 seconds then a message is signalled (te_send(34)).

Rule 35:
if the number of records requested in a TYPE command exceeds the threshold for the format requested, according to the table below, then a message is signalled (te_send(35)) and the command is cancelled.

FORMAT (record_format(r_last+1)	THRESHOLD (fmt_recs(record_format(r_last+1))
1	12
2	4
3	8
4	8
5	4
6	12
7	4
8	4

Rule 36:
if the argument of a SELECT command produces a null set and this argument falls within the bounds of a previous expand table (z_flag(c_last) = '1'b) then a message is signalled (te_send(36)) which gives the argument of the previous expand command.

DELETED
Rule 37:
if the number of dual repetitions(exact and essential simultaneously) thus far is less than 3(rep_knt_search(3)) and at least one such repetition occurs during this cycle(rep_knt_cycle(3)) then a message is signalled (te_send(37)) which gives the repeating commands.

Rule 38:
if the number of dual repetitions(exact and essential simultaneously) thus far is greater than 1 (rep_knt_search(3)) and at least one such repetition occurs during this cycle (rep_knt_cycle(3)) then a message is signalled (te_send(38)) which gives the repeating commands and help is offered.

Rule 39:
if the length of any string of commands equals 20 (string_length(s_last,st_last)) then a message is signalled (te_send(39)) and the user is logged off.

Rule 40:
if the number of consecutive command repetitions(cons_rep_knt) is 10 then a message is signalled (te_send(40)) and the user is logged off.

Rule 41:
if for at least 4 successive TYPE commands the average relevance of records viewed is increasing (fudge_rel + .25 < view_avg(r_last)) then a message is signalled (te_send(41)).

Rule 42:
if for at least 4 successive TYPE commands the average relevance of records viewed is static and no group had a relevance rating greater than or equal to 4 (fudge_rel_hi < 4) then a message is signalled (te_send(42)).

Rule 43:
if for at least 4 successive TYPE commands the average relevance of records viewed is static and at least one group had a relevance rating greater than or equal to 4 (fudge_rel_hi >= 4) then a message is signalled (te_send(43)) which gives the highly relevant group.

Rule 44:
if the argument of a SELECT command contains more than 3 words (space_knt) and a null set is generated (s_size (s_last)) then a message is signalled (te_send (44)).

Rule 45:
if the argument of a SELECT command contains more than 3 words (space_knt), there are fewer than 5 descriptors thus far in the entire search (d_last), and there are less than 4 records in the set (s_size (s_last)) then a message is signalled (te_send (45)).